Programozási

Verseny-feladatok

V.

Nemes Tihamér
Nemzetközi Informatikai Tanulmányi Verseny
2010-2014

Szerkesztette: Szlávi Péter, Zsakó László

A Nemes Tihamér NITV feladatsorait

Horváth Gyula (SzTE)
Zsakó László (ELTE)

vezetésével az NJSzT Országos Versenybizottsága mindenkori tagjai dolgozták ki:

Gulyás László (ELTE-AITIA), Horváth Győző (ELTE), Menyhárt László (ELTE), Szabadhegyi Csaba (ELTE), Szlávi Péter (ELTE).

Ezen túl az ELTE számos hallgatója vett részt a feladatok kidolgozásában és a verseny lebonyolításában.

ISBN 978-963-489-203-8

SZÉCHENYI 2020

Magyarország Kormánya

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Előszó

Magyarországon több kezdeti próbálkozás után 1985-ben szervezte meg az első országos középiskolai programozási versenyt Nemes Tihamér Országos Középiskolai Számítástechnikai Tanulmányi Verseny (NTOKSzTV) néven a Neumann János Számítógép-tudományi Társaság (NJSZT). A 9 évig kétfordulós versenyen 1989-ig egy korcsoportban, 1990-től külön korcsoportban indulhatnak a 14-15, illetve a 16-19 éves tanulók (azaz a középiskolák I.-II., illetve III.-V. osztályos diákjai). Az 1993/94-es tanévtől a versenyt három fordulóban rendezzük (az iskolai forduló és az országos döntő közé iktattunk egy regionális-megyei fordulót). A verseny első tíz helyezettje – a többi országos középiskolai tanulmányi versenyhez hasonlóan – érettségi, illetve felvételi kedvezményben részesül, és közülük válogatjuk ki az 1989 óta ugyancsak évente megrendezett Nemzetközi Informatikai Diákolimpia magyar résztvevőit is.

A Nemes Tihamér verseny, szerénytelenség nélkül megállapíthatjuk, népszerűvé vált a diákok körében: az utóbbi években 260-300 magyarországi középiskola kb. 2500-2500 I.-II., illetve III.-V. osztályos diákja vett részt a verseny első, iskolai fordulójában. Közülük kb. 600 diák jutott tovább a második fordulóba, majd kb. 180 a harmadikba. Említést érdemel, hogy a környező országokban élő, magyar anyanyelvű vagy magyarul jól beszélő diákok közül is egyre többen érdeklődnek a verseny iránt, illetve vesznek részt a versenyen; az Erdélyi Magyar Műszaki Tudományos Társaság szervezésében évente kb. 500 diák indul.

1991 óta különböző szervezési formákban az általános iskolások is részt vehetnek országos programozási jellegű versenyen. A határon túli résztvevők, valamint az általános iskolások megjelenése miatt a versenyt átkereszteltük, új neve: Nemes Tihamér Nemzetközi Informatikai Tanulmányi Verseny.

Sokak igényét elégítettük ki azzal, hogy a Nemes Tihamér NITV-t e korosztállyal bővítettük, a megrendezésre jelentkező megyéket (regionális versenybizottságokat) feladatokkal láttuk el, s megszerveztük az országos döntőt is.

Az NJSZT Országos Versenybizottsága sokak kívánságát teljesíti most azzal, hogy önálló kötetekben megjelenteti az eddigi versenyfeladatokat. Reméljük, hogy ezzel is segíteni tudjuk a leendő versenyzőket a felkészülésben, a tanáraikat pedig a számítástechnikai foglalkozások és szakkörök megtartásában.

Ebben a kötetben a 2010-2014 közötti Nemes Tihamér versenyek programozási feladatait ismertetjük. Minden egyes feladat után a javító tanároknak szóló megoldási és értékelési útmutatót is közöljük. A példatárban egyedülálló módon a feladatok részletes megoldását is közöljük, amelyek eddig még sehol nem jelentek meg. A versenyekre készülő diákoknak természetesen azt javasoljuk, hogy mielőtt a közölt megoldást és értékelést elolvasnák, saját maguk, önállóan próbálják megoldani a kitűzött feladatot. Nem törekedtünk szöveghűségre: ahol az eredeti megfogalmazást pontatlannak vagy hibásnak találtuk, módosítottunk a szövegen.

Tartalom

I.

Verseny-feladatok

Nemes Tihamér
Nemzetközi Informatikai Tanulmányi Verseny

2010. Első forduló

Ötödik-nyolcadik osztályosok

Számítógép nélküli feladatok

<u>1. feladat</u>: Számok (18 pont)

Az alábbi algoritmus 1 és 49 közötti számokat ír fel karakterekkel.

```
Szám(N):
   Ha N>39 akkor Ki: ”XL”; Egyes(N-40)
        különben Kettes(N)
Eljárás vége.

Kettes(N):
   Ciklus i=1-től (N div 10)-ig
      Ki: ”X”
   Ciklus vége
   Egyes(N mod 10)
Eljárás vége.

Egyes(N)
   Ha N=9 akkor Ki: ”IX”
   különben Ha N=4 akkor Ki: ”IV”
   különben Ha N>=5 akkor Ki: ”V”
            Ciklus i=1-től (N mod 5)-ig
               Ki: ”I”
            Ciklus vége
   Elágazás vége
Eljárás vége.
```

Mit ír ki a program az alábbi esetekben?

A. N=2 B. N=4 C. N=13 D. N=25 E. N=37 F. N=19
G. N=40 H. N=41 I. N=49

<u>2. feladat</u>: Testvérek (30 pont)

Az alábbi algoritmusban N rokoni kapcsolatot vizsgálunk. Az egyes embereket a sorszámukkal azonosítjuk. Az i-edik kapcsolatban A(i,1) jelenti a szülőt, A(i,2) pedig a gyereket. Az eljárás X és Y emberről mondaná meg, hogy ők milyen testvérek, ha jól működne. Tudjuk, hogy ismerjük mindegyikük mindkét szülőjét.

```
Eljárás(X,Y):
  i:=1; DB:=0
  Ciklus amíg i≤N és DB≤2
    Ha A(i,1)=X akkor DB:=DB+1; SX(DB):=A(i,2)
    i:=i+1
  Ciklus vége
  i:=1; DB:=1
  Ciklus amíg i<N és DB<2
    Ha A(i,1)=Y akkor SY(DB):=A(i,1); DB:=DB+1
    i:=i+1
  Ciklus vége
  Ha SX(1)=SY(1) és SX(2)=SY(2) akkor Ki: ”Testvérek”
  különben ha SX(1)=SY(1) vagy SX(2)=SY(2)
              akkor Ki: ”Testvérek”
           különben Ki: ”Nem testvérek”
Eljárás vége.
```

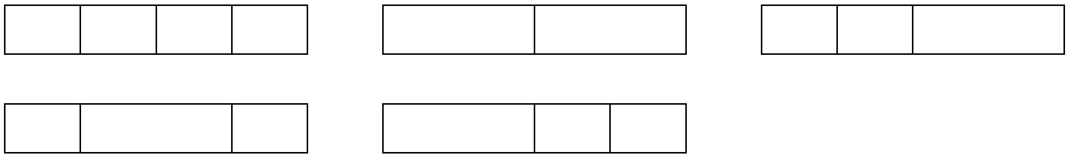
A. Az algoritmusban több hiba is van, melyek ezek?

B. Az első ciklus gyorsabbá tehető a ciklusfeltétel módosításával. Hogyan?

<u>3. feladat</u>: Járda (20 pont)

Egy N egység méretű járdát 1 és 2 méretű lapokkal szeretnénk kikövezni.

<u>Példa</u>: N=4 egység hosszú járda kikövezési lehetőségei:



A. Add meg, hányféleképpen lehet kikövezni az N=1, N=2, N=3 hosszú járdákat!

B. Rajzold le az összes lehetséges kikövezést N=3 esetén!

C. Add meg, hányféleképpen lehet kikövezni az N=5 egység hosszú járdát!

D. Add meg, hányféleképpen lehet kikövezni az N=6 egység hosszú járdát!

E. Add meg, hányféleképpen lehet kikövezni az N=10 méretű járdát!

Számítógépes feladat – VÁLASZTHATÓ

<u>4. feladat</u>: Eső (30 pont)

N napon át minden nap megmértük, hogy hány milliméter eső esett (N≥0). Készíts programot, amely megadja, hogy

A. Mennyi eső esett összesen az N nap alatt?

B. Melyik napon esett a legtöbb eső?

C. Hány esős nap volt?

<u>Példa</u>:

```
N=5, mérések: 0,15,20,0,10
Az összes eső: 45 milliméter
A legesősebb nap: 3
Esős napok száma: 3
```

Számítógép nélküli feladat – VÁLASZTHATÓ

<u>4. feladat</u>: Kannák (30 pont)

Egy gazdának két kannája van, az egyik 4, a másik 9 literes. Adott mennyiségű vizet szeretne ki-mérni. A mérés során a következő műveleteket tudja végezni:

A. A 9-literes kanna teletöltése

B. A 4-literes kanna teletöltése

C. A 9-literes kanna kiürítése

D. A 4-literes kanna kiürítése

E. Áttöltés a 9-literesből a 4-literesbe (amíg tele nem lesz, ill. van benne)

F. Áttöltés a 4-literesből a 9-literesbe (amíg tele nem lesz, ill. van benne)

Adj olyan legrövidebb műveletsort, amelynek végén valamelyik kannában

A. 1 liter,

B. 3 liter,

C. 6 liter

D. 5 liter

E. 8 liter

víz keletkezik.

Pl. 5 liter kimérése az AE műveletsorral lehetséges.

Kilencedik-tizedik osztályosok

1. feladat: Számok (20 pont)

Az alábbi algoritmus 1 és 100 közötti számokat ír fel karakterekkel.

```
Számok(N):
  Ha N=100 akkor Ki:”C”
  különben Ha N>89 akkor Ki: ”XC”; Egyes(N-90)
  különben ha N>=50 akkor Ki: ”L”; Kettes(N-50)
  különben Ha N>39 akkor Ki: ”XL”; Egyes(N-40)
  különben Kettes(N)
Eljárás vége.
```

```
Kettes(N):
  Ciklus i=1-től (N div 10)-ig
    Ki: ”X”
  Ciklus vége
  Egyes(N mod 10)
Eljárás vége.
```

```
Egyes(N)
  Ha N=9 akkor Ki: ”IX”
  különben Ha N=4 akkor Ki: ”IV”
  különben Ha N>=5 akkor Ki: ”V”
           Ciklus i=1-től (N mod 5)-ig
             Ki: ”I”
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

Mit ír ki a program az alábbi esetekben?

A. N=2	B. N=4	C. N=13	D. N=25	E. N=99	F. N=19
G. N=40	H. N=41	I. N=49	J. N=77	K. N=64	L. N=44

2. feladat: Testvérek (24 pont)

Az alábbi algoritmusban N rokoni kapcsolatot vizsgálunk. Az egyes embereket a sorszámukkal azonosítjuk. Az i-edik kapcsolatban A(i,1) jelenti a szülőt, A(i,2) pedig a gyereket. Az eljárás X testvéreit és féltestvéreit adná meg, ha jól működne.

```
Eljárás(X):
  i:=1; DB:=0
  Ciklus amíg i<N és DB<2
    Ha A(i,2)=X akkor DB:=DB+1; SX(DB):=A(i,1)
    i:=i+1
  Ciklus vége
```

```
  D:=0
  Ciklus i=1-től N-ig
    Ha A(i,1)=SX(1) vagy A(i,1)=SX(2)
       akkor D:=D+1; T(DB):=A(DB,2)
  Ciklus vége
Eljárás vége.
```

A. Az algoritmusban több elírásból származó hiba is van, melyek ezek?

B. Az algoritmusban több elvi hiba is van, ami akkor is baj lenne, ha nem követtünk volna el elírási hibákat, melyek ezek?

3. feladat: Mit csinál? (21 pont)

Egy internetes játékban a szervezők N napon feljegyezték, hogy kik léptek be. Ha valaki egy nap többször is bejelentkezett, akkor azt is számolták. Összesen M játékos vett részt a játékban, de nem biztos, hogy az N nap alatt mindegyik belépett.

DB(i) jelöli azt, hogy az i-edik napon hány játékos lépett be, AZ(i,j) az i-edik nap j-edik belépőjének azonosítója, BDB(i,j) pedig az aznapi belépéseinek száma. Az alábbi algoritmusok ezek alapján számolnak ki valamit:

```
Első(D,E):
  D:=0
  Ciklus i=1-től N-ig
    Ciklus j=1-től DB(i)-ig
      k:=1
      Ciklus amíg k≤D és AZ(i,j)≠E(k)
        k:=k+1
      Ciklus vége
      Ha k>D akkor D:=D+1; E(D):=AZ(i,j)
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

```
Második(D,E,F):
  D:=0
  Ciklus i=1-től N-ig
    Ciklus j=1-től DB(i)-ig
      k:=1
      Ciklus amíg k≤D és AZ(i,j)≠E(k)
        k:=k+1
      Ciklus vége
      Ha k>D akkor D:=D+1; E(D):=AZ(i,j); F(D):=BDB(i,j)
          különben ha F(k)>BDB(i,j) akkor F(k):=BDB(i,j)
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

```
Harmadik(H):
  D:=0; G:=1; H:=E(1)
  Ciklus i=1-től N-ig
    Ciklus j=1-től DB(i)-ig
      k:=1
      Ciklus amíg k≤D és AZ(i,j)≠E(k)
        k:=k+1
      Ciklus vége
      Ha k>D akkor D:=D+1; E(D):=AZ(i,j); F(D):=1
          különben F(k):=F(k)+1
      Ha F(k)>F(G) akkor G:=k; H:=E(k)
    Ciklus vége
  Ciklus vége
  H:=E(G)
Eljárás vége.
```

<u>Példa</u>:

DB	AZ				BDB			
3	Alfa	Béta	Gamma		2	15	1	
2	Gamma	Alfa			1	4		
4	Gamma	Alfa	Delta	Béta	1	1	6	1
3	Gamma	Alfa	Béta		1	3	2	
3	Gamma	Béta	Alfa		1	3	1	

A. Mit adnak eredményül a fenti eljárások a táblázatban szereplő értékek esetén?

B. Fogalmazd meg általánosan, mi az egyes eljárások eredménye!

<u>4. feladat</u>: Kannák (15 pont)

Egy gazdának két kannája van, az egyik 4, a másik 9 literes. Adott mennyiségű vizet szeretne kimérni. A mérés során a következő műveleteket tudja végezni:

A. A 9-literes kanna teletöltése
B. A 4-literes kanna teletöltése
C. A 9-literes kanna kiürítése
D. A 4-literes kanna kiürítése
E. Áttöltés a 9-literesből a 4-literesbe (amíg tele nem lesz, ill. van benne)
F. Áttöltés a 4-literesből a 9-literesbe (amíg tele nem lesz, ill. van benne)

Adj olyan legrövidebb műveletsort, amelynek végén valamelyik kannában

A. 1 liter, B. 3 liter, C. 6 liter D. 2 liter E. 7 liter

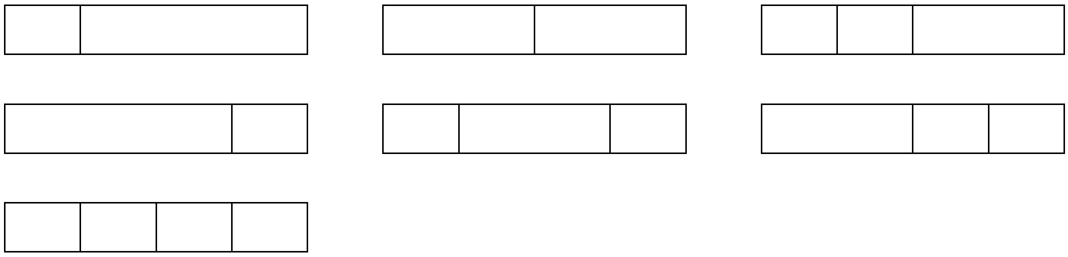
víz keletkezik.

Pl. 8 liter kimérése a BFBF műveletsorral lehetséges.

<u>5. feladat</u>: Járda (20 pont)

Egy N egység méretű járdát 1, 2 és 3 méretű lapokkal szeretnénk kikövezni.

<u>Példa</u>: N=4 egység hosszú járda kikövezési lehetőségei:



A. Add meg, hányféleképpen lehet kikövezni az N=1, N=2, N=3 hosszú járdákat!

B. Rajzold le az összes lehetséges kikövezést N=3 esetén!

C. Add meg, hányféleképpen lehet kikövezni az N=5 egység hosszú járdát!

D. Add meg, hányféleképpen lehet kikövezni az N=6 egység hosszú járdát!

E. Add meg, hányféleképpen lehet kikövezni az N=10 méretű járdát!

Tizenegyedik-tizenharmadik osztályosok

<u>1. feladat</u>: Számok (20 pont)

Az alábbi algoritmus 1 és 1000 közötti számokat ír fel karakterekkel.

```
Római(N):
  Ha N=1000 akkor Ki: ”M”
  különben
     Ha N>899 akkor Ki: ”CM”; N:=N-900
     különben ha N>=500 akkor Ki: ”D”; N:=N-500
     különben Ha N>399 akkor Ki: ”CD”; N:=N-400
     Hármas(N)
  Elágazás vége
Eljárás vége.

Hármas(N):
  Ciklus i=1-től (N div 100)-ig
    Ki: ”C”
  Ciklus vége
  N:=N mod 100
  Ha N>89 akkor Ki: ”XC”; Egyes(N-90)
  különben ha N>=50 akkor Ki: ”L”; Kettes(N-50)
  különben Ha N>39 akkor Ki: ”XL”; Egyes(N-40)
  különben Kettes(N)
Eljárás vége.

Kettes(N):
  Ciklus i=1-től (N div 10)-ig
    Ki: ”X”
  Ciklus vége
  Egyes(N mod 10)
Eljárás vége.
```

```
Egyes(N)
  Ha N=9 akkor Ki: ″IX″
  különben Ha N=4 akkor Ki: ″IV″
  különben Ha N>=5 akkor Ki: ″V″
           Ciklus i=1-től (N mod 5)-ig
             Ki: ″I″
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

Mit ír ki a program az alábbi esetekben?

A. N=2	B. N=14	C. N=43	D. N=85	E. N=99	F. N=123
G. N=340	H. N=451	I. 499	J. N=877	K. N=619	L. N=999

2. feladat: Testvérek (22 pont)

Az alábbi algoritmusban N rokoni kapcsolatot vizsgálunk. Az egyes embereket a sorszámukkal azonosítjuk. Az i-edik kapcsolatban A(i,1) jelenti a szülőt, A(i,2) pedig a gyereket. Az eljárás azokat adná meg, akikkel X-nek közös gyereke van, ha jól működne.

```
Eljárás(X):
  i:=1; DB:=0
  Ciklus amíg i<N
    Ha A(i,2)=X akkor DB:=DB+1; SX(DB):=A(DB,1)
    i:=i+1
  Ciklus vége
  D:=0
  Ciklus i=1-től N-ig
    j:=1
    Ciklus amíg j≤DB és A(i,2)=SX(i)
      j:=j+1
    Ciklus vége
    Ha j>DB akkor D:=D+1; T(DB):=A(i,1)
  Ciklus vége
Eljárás vége.
```

A. Az algoritmusban több elírásból származó hiba is van, melyek ezek?

B. Az algoritmusban több elvi hiba is van, ami akkor is baj lenne, ha nem követtünk volna el elírási hibákat, melyek ezek?

3. feladat: Mit csinál? (22 pont)

Egy internetes játékban a szervezők N napon feljegyezték, hogy kik léptek be. Ha valaki egy nap többször is bejelentkezett, akkor azt is számolták. Összesen M játékos vett részt a játékban, de nem biztos, hogy az N nap alatt mindegyik belépett.

DB(i) jelöli azt, hogy az i-edik napon hány játékos lépett be, AZ(i,j) az i-edik nap j-edik belépőjének azonosítója, BDB(i,j) pedig az aznapi belépéseinek száma. Az alábbi algoritmusok ezek alapján számolnak ki valamit:

```
Egy(D,E):
  Ciklus j=1-től DB(1)-ig
    E(j):=AZ(1,j)
  Ciklus vége
  D:=DB(1)
  Ciklus i=2-től N-ig
    Ciklus k=1-től D-ig
      j:=1
      Ciklus amíg j≤DB(i) és AZ(i,j)≠E(k)
        j:=j+1
      Ciklus vége
      Ha j>DB(i) akkor E(k):=E(D); D:=D-1
    Ciklus vége
  Ciklus vége
Eljárás vége.
Kettő(D,E,F):
  D:=0
  Ciklus i=1-től N-ig
    Ciklus j=1-től DB(i)-ig
      k:=1
      Ciklus amíg k≤D és AZ(i,j)≠E(k)
        k:=k+1
      Ciklus vége
      Ha k>D akkor D:=D+1; E(D):=AZ(i,j); F(D):=BDB(i,j)
          különben ha F(k)<BDB(i,j) akkor F(k):=BDB(i,j)
    Ciklus vége
  Ciklus vége
Eljárás vége.
Három(H):
  D:=0; G:=1; H:=E(1)
  Ciklus i=1-től N-ig
    Ciklus j=1-től DB(i)-ig
      k:=1
      Ciklus amíg k≤D és AZ(i,j)≠E(k)
        k:=k+1
      Ciklus vége
      Ha k>D akkor D:=D+1; E(D):=AZ(i,j); F(D):=BDB(i,j)
          különben F(k):=F(k)+BDB(i,j)
      Ha F(k)>F(G) akkor G:=k; H:=E(k)
    Ciklus vége
  Ciklus vége
  H:=E(G)
Eljárás vége.
```

Példa:

DB	AZ				BDB			
3	Alfa	Béta	Gamma		2	15	1	
2	Gamma	Alfa			1	4		
4	Gamma	Alfa	Delta	Béta	1	1	6	1
3	Gamma	Alfa	Béta		1	3	2	
3	Gamma	Béta	Alfa		1	3	1	

A. Mit adnak eredményül a fenti eljárások a táblázatban szereplő értékek esetén?

B. Fogalmazd meg általánosan, mi az egyes eljárások eredménye!

<u>4. feladat</u>: Kannák (15 pont)

Egy gazdának két kannája van, az egyik 4, a másik 9 literes. Adott mennyiségű vizet szeretne kimérni. A mérés során a következő műveleteket tudja végezni:

A. A 9-literes kanna teletöltése
B. A 4-literes kanna teletöltése
C. A 9-literes kanna kiürítése
D. A 4-literes kanna kiürítése
E. Áttöltés a 9-literesből a 4-literesbe (amíg tele nem lesz, ill. van benne)
F. Áttöltés a 4-literesből a 9-literesbe (amíg tele nem lesz, ill. van benne)

Adj olyan legrövidebb műveletsort, amelynek végén valamelyik kannában

A. 1 liter   B. 3 liter   C. 6 liter   D. 2 liter   E. 7 liter

víz keletkezik!

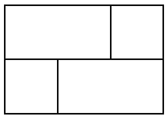
Pl. 8 liter kimérése a BFBF műveletsorral lehetséges.

<u>5. feladat</u>: Járdakövezés (18 pont)

Egy 2*N egység méretű járdát 1*1 és 1*2 méretű lapokkal szeretnénk kikövezni. Hányféleképpen lehet ezt megtenni?

<u>Példa</u>: N=2 egység hosszú járda kikövezési lehetőségeinek száma 7.

<u>Példa</u>: Egy N=3 hosszú járdát például így is ki lehet kövezni:



A. Rajzold le az összes lehetséges kikövezést N=2 esetén!
B. Add meg hányféleképpen lehet kikövezni az N=3 hosszú járdát!
C. Add meg hányféleképpen lehet kikövezni az N=4 hosszú járdát!
D. Add meg hányféleképpen lehet kikövezni az N=5 hosszú járdát!
E. Add meg hányféleképpen lehet kikövezni az N=7 hosszú járdát!
F. Add meg hányféleképpen lehet kikövezni az N=8 hosszú járdát!

2010. Második forduló

Ötödik-nyolcadik osztályosok

1. feladat: Kártya (25 pont)

Egy kártyajátékban az egyes lapoknak számértékük van. Minden lapot egy színnel és egy figurával adunk meg. A színek: piros, zöld, tök, makk. A figurák: 7-es, 8-as, 9-es, 10-es, alsó, felső, király, ász. A számot tartalmazó figurák annyit érnek, amennyi a ráírt szám. Az alsó 2-t, a felső 3-at, a király 4-et, az ász 11-et ér. A piros lapoknál az értéket duplán kell számítani.

Készíts programot, amely beolvassa egy játékos kártyáit ($1 \le N \le 4$), majd megadja, hogy a lapok összesen hány pontot érnek!

Példa

```
Bemenet:
Kártyák száma? 3
1. kártya színe? piros
1. kártya figurája? alsó
2. kártya színe? tök
2. kártya figurája? 7-es
3. kártya színe? tök
3. kártya figurája? ász
```

```
Kimenet:
A kártyák értéke: 22
```

2. feladat: Bábu (25 pont)

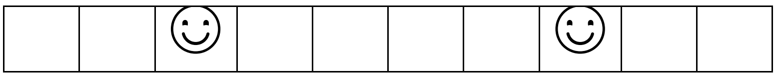
Egy játéktáblán a 0. időegységben L bábu van. Mindegyiket elindítjuk valamerre. Egy időegység alatt mindegyik a neki megfelelő távolságra mozdul el, a tábla széléről visszafordulnak. Lehetséges, hogy előbb-utóbb két bábu összeütközik: ugyanarra a helyre lépnének vagy átlépnének egymáson.

Írj programot, amely megadja, hogy K időegységen belül mikor ütközik legelőször két bábu!

A program olvassa be a tábla szélességét ($1 \le N \le 100$), a bábuk számát ($1 \le L \le 10$) és az időtartamot ($1 \le K \le 100\,000$)! Ezután olvassa be a bábuk kezdő helyét ($1 \le S_i \le N$) és mozgás irányát ($X_i \in \{J, B\}$ – jobbra, balra)!

A program írja ki az első ütközés időpontját! Ha K időegységen belül nincs ütközés, akkor -1-et kell kiírni!

Példa:



```
A tábla hossza: 10
A bábuk száma: 2
Az időtartam: 10
1 bábu helye: 3, iránya: B
2. bábu helye: 8, iránya: B
Ütközés időpont: 5
```

Magyarázat: A bábuk helyzete időegységenként:

```
1: 2, 7
2: 1, 6
3: 2, 5
4: 3, 4
5: 4, 3
```

Az 5. időegységre az 1-es és a 2-es bábu egymáson átlépett volna, azaz az 5. időegységben ütköztek.

3. feladat: Járdakövezés (25 pont)

Egy N egység hosszú járdát (1≤N≤80) 1 és 2 méretű lapokkal szeretnénk kikövezni. Hányféleképpen lehet ezt megtenni?

Készíts programot, amely kiszámítja, hogy egy N egység hosszú járdát hányféleképpen lehet kikövezni 1 és 2 méretű lapokkal!

Példa: N=4 egység hosszú járdát 5-féleképpen lehet kikövezni, a kikövezési lehetőségei:

```
Bemenet:                   Kimenet:
A járda hossza? 4          A kikövezések száma: 5
```

Kilencedik-tizedik osztályosok

1. feladat: Kép (20 pont)

Ugyanarról a területről két időpontban készítettünk fényképet. A fényképek négy széléről le szeretnénk vágni azt a részt, amelyek egyformák.

Készíts programot, amely megadja, hogy a kép 4 széléről maximum mekkora téglalapok vághatók le!

A bemenet első sorában a fényképek sorai és oszlopai száma van (1≤N,M≤1000). A következő N sorban az első kép, az azt követő N sorban a második kép képpontjai vannak. Minden sor M képpont leírását tartalmazza, egymástól egy-egy szóközzel elválasztva. A képpontokat egy 0 és 255 közötti fényességértékkel adjuk meg.

A kimenet első sorába a legnagyobb balról, alulról, jobbról, illetve felülről levágható téglalap szélességét kell írni!

Példa:

```
Bemenet:                   Kimenet:
8 10                       1 1 3 2
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 5 5 5 5
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
-------------------
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 3 3 3 3 3
2 2 9 9 2 2 2 2 2 2
2 2 2 2 2 2 5 5 5 5
1 1 1 1 1 1 1 1 1 1
1 3 1 1 3 1 1 1 1 1
1 1 1 1 1 1 5 1 1 1
0 0 0 0 0 0 0 0 0 0
```

2. feladat: Játék (20 pont)

Egy játéktáblán a 0. időegységben L bábu van. Mindegyiket elindítjuk valamerre. Egy időegység alatt mindegyik a neki megfelelő távolságra mozdul el, a tábla szélére érve megállnak. Lehetséges, hogy előbb-utóbb két bábu összeütközik: ugyanarra a helyre lépnének vagy átlépnének egymáson.

Írj programot, amely megadja, hogy K időegységen belül mikor ütközik legelőször két bábu!

A bemenet első sorában a játéktábla sorai és oszlopai száma ($1 \leq N, M \leq 100$), a bábuk száma ($1 \leq L \leq 10$) és az időtartam ($1 \leq K \leq 100\,000$) van. A következő L sor egy-egy bábu leírását tartalmazza: a kezdő helyét ($1 \leq S_i \leq N$, $1 \leq O_i \leq M$) és mozgás irányát ($X_i \in \{F, L, J, B\}$ – fel, le, jobbra, balra).

A kimenet egyetlen sorába az első ütközés időpontját kell írni! Ha K időegységen belül nincs ütközés, akkor -1-et kell kiírni!

Példa:

Bemenet:	Kimenet:
7 10 3 100 4 3 J 2 6 F 4 8 B	3

Magyarázat: A bábuk helyzete időegységenként:

```
1: (4,4), (1,6), (4,7)
2: (4,5), (1,6), (4,6)
3: (4,6), (1,6), (4,5)
```

A 3. időegységre az 1-es és a 3-as bábu egymáson átlépett volna, azaz a 3. időegységben ütköztek.

3. feladat: Ütemezés (20 pont)

Egy vállalkozó alkatrészek gyártásával foglalkozik. Minden alkatrészen kétféle műveletet kell elvégeznie, A és B műveletet. Mindkét művelet elvégzésére egy-egy munkagépe van, amelyek egymástól függetlenül tudnak dolgozni, de egy alkatrészen egyszerre csak egyik művelet végezhető. Az alkatrészen a két műveletet tetszőleges sorrendben el lehet végezni. Minden legyártandó alkatrészre ismert, hogy mennyi időt igényel az A, valamint a B művelet elvégzése.

Készíts programot, amely kiszámítja, hogy legkevesebb mennyi idő alatt lehet legyártani az összes alkatrészt!

A bemenet első sorában az alkatrészek száma ($2 \leq N \leq 1000$) van. A második és a harmadik sor pontosan N egész számot tartalmaz, a legyártandó alkatrészeken elvégzendő A, illetve B műveletek idejét. A második sorban az i-edik szám az i-edik alkatrészen végzendő A művelet ideje. A harmadik sorban az i-edik szám pedig az i-edik alkatrészen végzendő B művelet ideje. A második és harmadik sorban lévő számok mindegyike 1 és 5000 közötti érték.

A kimenet első sora egy egész számot tartalmazzon, azt a legkisebb T időt, amely alatt a két gép le tudja gyártani az összes alkatrészt!

Példa:

Bemenet:	Kimenet:
3 4 20 15 11 30 3	50

4. feladat: Járdakövezés (15 pont)

Egy N egység hosszú járdát 1, 2 és 3 méretű lapokkal szeretnénk kikövezni. Hányféleképpen lehet ezt megtenni?

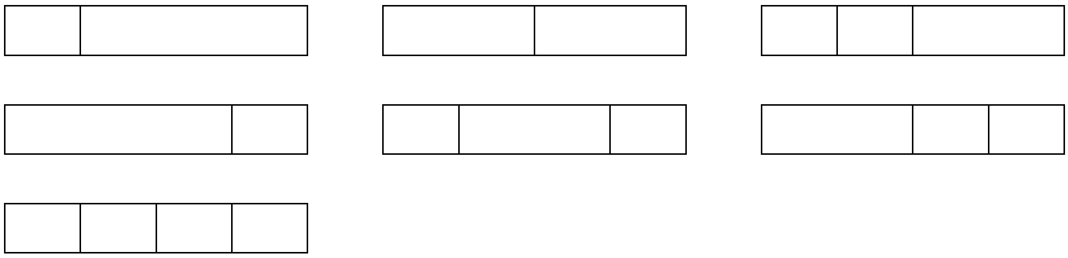
Készíts programot, amely kiszámítja, hogy egy N egység hosszú járdát hányféleképpen lehet kikövezni 1, 2 és 3 méretű lapokkal!

A bemenet egyetlen sorában a járda hossza (`1≤N≤70`) van.

A kimenet egyetlen sora a lehetséges kikövezések számát tartalmazza!

Példa:

Bemenet:	Kimenet:
4	7



Tizenegyedik-tizenharmadik osztályosok

1. feladat: Műhold (15 pont)

Egy műhold ugyanarról a területről két időpontban készített fényképet. A két fényképen különbségek találhatók.

Készíts programot, amely megadja azt a legkisebb téglalapot, amelyen kívül a két fénykép teljesen azonos!

A bemenet első sorában a fényképek sorai és oszlopai száma van (`1≤N,M≤1000`). A következő N sorban az első kép, az azt követő N sorban a második kép képpontjai vannak. Minden sor M képpont leírását tartalmazza. A képpontokat egy 0 és 255 közötti fényességértékkel adjuk meg.

A kimenet első sorába a legkisebb olyan téglalap bal felső sarkának és jobb alsó sarkának sor- és oszlopindexeit kell írni, amelyen kívül a két kép teljesen megegyezik!

Példa:

Bemenet:
```
8 10
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 5 5
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
-------------------
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 3 3 3 3 3
2 2 9 9 2 2 2 2 2 2
2 2 2 2 2 2 2 5 5 5
1 1 1 1 1 1 1 1 1 1
```

Kimenet:
```
3 2 7 8
```

```
1 3 1 1 3 1 1 1 1 1
1 1 1 1 1 1 5 1 1 1
0 0 0 0 0 0 0 0 0 0
```

2. feladat: Játék (15 pont)

Egy játéktáblán a 0. időegységben L bábu van. Mindegyiket elindítjuk valamerre. Egy időegység alatt mindegyik a neki megfelelő irányba mozdul el, a tábla szélén mozgás irányukat az ellenkezőre változtatják. Lehetséges, hogy előbb-utóbb két bábu összeütközik: ugyanarra a helyre lépnének vagy átlépnének egymáson.

Írj programot, amely megadja, hogy K időegységen belül mikor ütközik legelőször két bábu!

A bemenet első sorában a játéktábla sorai és oszlopai száma (1≤N,M≤100), a bábuk száma (1≤L≤10) és az időtartam (1≤K≤100 000) van. A következő L sor egy-egy bábu leírását tartalmazza: a kezdő helyét ($1≤S_i≤N$, $1≤O_i≤M$) és mozgás irányát ($X_i \in$ {F,L,J,B} – fel, le, jobbra, balra).

A kimenet egyetlen sorába az első ütközés időpontját kell írni! Ha K időegységen belül nincs ütközés, akkor -1-et kell kiírni!

Példa:

Bemenet:	Kimenet:
7 10 3 100 4 3 J 2 6 F 4 8 B	3

Magyarázat: A bábuk helyzete időegységenként:

```
1: (4,4), (1,6), (4,7)
2: (4,5), (2,6), (4,6)
3: (4,6), (3,6), (4,5)
```

A 3. időegységre az 1-es és a 3-as bábu egymáson átlépett volna, azaz a 3. időegységben ütköztek.

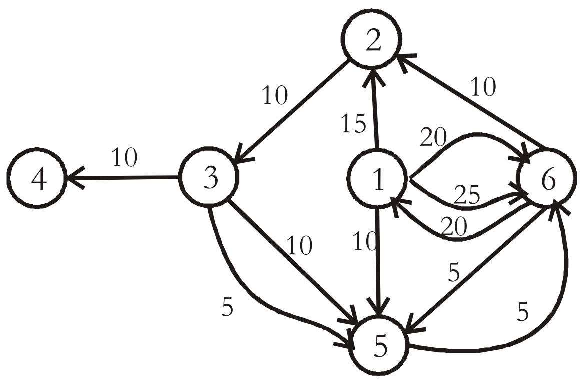
3. feladat: Futár (15 pont)

Egy országban N város van. Az egyik városban futárok várakoznak. Ismerjük azt is, hogy az egyes városokból mennyi idő alatt lehet elérni valamilyen járművel adott másik városokat.

Készíts programot, amely megadja, hogy minimum mennyi idő alatt jutnak el a futárok az összes városba!

A bemenet első sorában a városok száma (1≤N≤100), a járművek száma (1≤M≤5000) és a futárok kezdő helye (1≤H≤N) van. A következő M sorban szerepel egy-egy járművet leíró indulás helye ($1≤A_i≤N$), érkezés helye ($1≤B_i≤N$), valamint az út megtételéhez szükséges idő ($1≤T_i≤1000$).

A kimenetre egyetlen sort kell írni, a minimális időt, ami alatt a futárok az összes városba elérhetnek!



Példa:

Bemenet:	Kimenet:
6 12 1 1 6 20 1 5 10 5 6 5 6 2 10 1 2 15 2 3 10 3 4 10 3 5 10 1 6 25 3 5 5 6 1 20 6 5 5	35 Magyarázat: 1 → 5: 10 időegység 1 → 5 → 6: 15 időegység 1 → 2: 15 időegység 1 → 2 → 3: 25 időegység 1 → 2 → 3 → 4: 35 időegység

4. feladat: Ütemezés (15 pont)

Egy vállalkozó alkatrészek gyártásával foglalkozik. Minden alkatrészen kétféle műveletet kell elvégeznie, A és B műveletet. Mindkét művelet elvégzésére egy-egy munkagépe van, amelyek egymástól függetlenül tudnak dolgozni. Minden alkatrészen először az A műveletet kell elvégezni, majd ezután lehet elvégezni a B műveletet (bármikor, nem feltétlenül folyamatosan). Minden legyártandó alkatrészre ismert, hogy mennyi időt igényel az A, valamint a B művelet elvégzése.

Készíts programot, amely kiszámítja, hogy legkevesebb mennyi idő alatt lehet legyártani az összes alkatrészt!

A bemenet első sorában az alkatrészek száma van (2≤N≤2000). A második és a harmadik sor pontosan N számot tartalmaz, a legyártandó alkatrészeken elvégzendő A, illetve B műveletek idejét. A második sorban az i-edik szám az i-edik alkatrészen végzendő A művelet ideje. A harmadik sorban az i-edik szám pedig az i-edik alkatrészen végzendő B művelet ideje. A második és harmadik sorban lévő számok mindegyike 1 és 50 közötti érték.

A kimenet első sora egy egész számot tartalmazzon, azt a legkisebb T időt, amely alatt a két gép le tudja gyártani az összes alkatrészt! A második sor az alkatrészek sorszámát tartalmazza abban a sorrendben, ahogy azokon az A műveletet kell elvégezni! A harmadik sor az alkatrészek sorszámát tartalmazza abban a sorrendben, ahogy azokon az B műveletet kell elvégezni! Több megoldás esetén bármelyik megadható.

Példa:

Bemenet:	Kimenet:
3 8 1 6 1 6 3	16 2 3 1 2 3 1

5. feladat: Járdakövezés (15 pont)

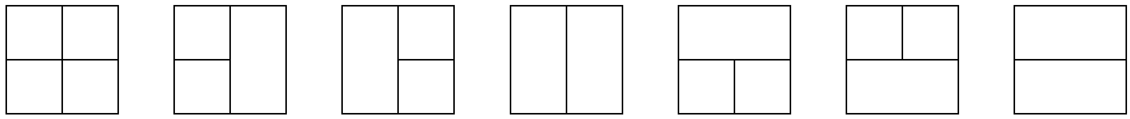
Írj programot, amely kiszámítja, hogy hány féleképpen lehet kikövezni egy 2*N egység méretű járdát 1*1 és 1*2 méretű lapokkal!

A bemenet első sorában a járda méretét megadó egész szám van (`1≤N≤36`).

A kimenet első és egyetlen sorába azt a számot kell írni, ahányféleképpen kikövezhető a 2*N méretű járda!

Példa:

Bemenet:	Kimenet:
2	7



2010. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Lövészverseny (25 pont)

Egy lövészversenyen a versenyzők egymás után lőnek. Ismerjük N (1≤N≤1000) versenyző eredményét. Készíts programot, amely beolvassa N értékét és az N db eredményt, majd megadja:

A. minden versenyzőre, hogy az addig szereplők közül hányan értek el nála jobb eredményt;

B. azokat a versenyzőket, akik a verseny valamelyik időszakában álltak az első helyen;

C. azokat a versenyzőket, akik a verseny valamelyik időszakában álltak az utolsó helyen;

D. a verseny győztesét!

Példa:

```
Bemenet:
N=6
1. versenyző: 594
2. versenyző: 596
3. versenyző: 582
4. versenyző: 599
5. versenyző: 590
6. versenyző: 590
```

```
Kimenet:
Jobb eredmény: 0 0 2 0 3 3
Állt az első helyen: 1 2 4
Állt az utolsó helyen: 1 3
Győztesek: 4
```

2. feladat: Tördelés (25 pont)

Készíts programot, amely egy beolvasott szöveget (legfeljebb 1000 karakteres) a képernyő S (10≤S≤80) szélességű tartományába tördel úgy sorokra, hogy szavak nem törhetnek két sorba!

Példa:

```
Bemenet:
S=20
Szöveg=
"Készíts programot, amely egy beolvasott szöveget a képernyő S
szélességű tartományába tördel úgy sorokra, hogy szavak nem
törhetnek két sorba!"
```

Kimenet: (az ábrán függőleges vonal jelöli a margókat, ezt a programnak nem kell kirajzolnia)

```
|Készíts programot,   |
|amely egy beolvasott|
|szöveget a képernyő |
|S szélességű        |
|tartományába tördel |
|úgy sorokra, hogy   |
|szavak nem törhetnek|
|két sorba!          |
```

3. feladat: Verseny (25 pont)

Egy pontozásos versenyen N (1≤N≤100) résztvevő indul. A versenyzőket M (3≤M≤10) pontozó pontozza. A pontozók azonban befolyásolhatnák a verseny végeredményét (a saját országbeli versenyzőknek nagyobb, a más országbeli versenyzőknek pedig kisebb pontot adva, ezért a verseny szervezői úgy döntöttek, hogy az M pontszám közül a legkisebbet és a legnagyobbat egy versenyzőnél sem veszik figyelembe, azaz mindenkit M-2 pontszám összegével értékelnek.

Készíts programot, amely

A. kiírja minden versenyzőre, hogy mely pontozók pontszámát hagyják ki;

B. kiszámítja és kiírja minden versenyző pontszámát;

C. kiírja a verseny végeredményét (a versenyzőket pontszám szerint csökkenő sorrendben, holtverseny esetén azonos helyezési számmal)!

Példa:

```
Bemenet:
N=4, M=4
1. versenyző pont: 8 2 6 6
2. versenyző pont: 5 5 5 5
3. versenyző pont: 4 5 6 7
4. versenyző pont: 5 4 6 5
```

```
Kimenet:
Kihagyott pontozók:
1. versenyző: 1 2
2. versenyző: 1 2
3. versenyző: 1 4
4. versenyző: 2 3
Pontszámok:
1. versenyző: 12 pont
2. versenyző: 10 pont
3. versenyző: 11 pont
4. versenyző: 10 pont
Sorrend:
1. helyezett: 1. versenyző 12 pont
2. helyezett: 3. versenyző 11 pont
3. helyezett: 2. versenyző 10 pont
3. helyezett: 4. versenyző 10 pont
```

Kilencedik-tizedik osztályosok

1. feladat: Játék (19 pont)

Egy N napos játék-versenyen, ahol nem kötelező minden nap játszani, 3 játékos (A,B,C) vesz részt. Nem szerencsések azok a napok, amikor pontosan 2 játékos vesz részt a versenyen, mert ekkor közös stratégiát alkothatnak a harmadik ellen.

Készíts programot, amely megadja, hogy mikor voltak ilyen helyzetek!

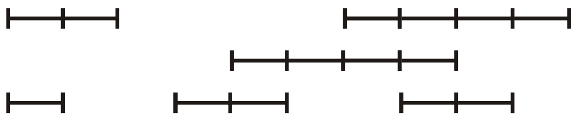
A bemenet első sorában a napok száma van ($1 \le N \le 10\,000$). A második sorban az A, a harmadikban a B, a negyedikben pedig a C játékos leírása van. Mindhárom sor első száma ($1 \le M \le N$) azt jelenti, hogy a játékos a verseny hány szakaszában vett rész. Ezt M számpár követi: az első tagja a szakasz első napjának sorszáma, a második pedig a szakasz hossza.

A kimenet első sorába azon intervallumokat kell írni, amikor A és B, a második sorba azokat, amikor A és C, a harmadikba pedig azokat, amikor B és C játékos volt kettesben! Minden sor az ilyen intervallumok számával kezdődjön, majd az intervallumok kezdete és vége kövesse, kezdet szerint növekvő sorrendben!

Példa:

```
Bemenet:
10
2 1 2 7 4
1 5 4
3 1 1 4 2 8 2
```

```
Kimenet:
1 7 7
2 1 1 9 9
1 5 5
```



2. feladat: Térkép (16 pont)

Ismerjük egy N*M-es térképen a pontok tengerszint feletti magasságát. Két szomszédos pont közötti lépés ideje 1 + a magasságuk különbségének abszolút értéke. Egy pontból nem lehet átlépni a szomszédjába, ha a magasságuk különbsége nagyobb, mint H. Egy P pontból szeretnénk eljutni egy Q pontba.

Készíts programot, amely megadja, hogy P-ből Q-ba minimálisan mennyi idő alatt lehet eljutni és megad egy ilyen utat az L,J,F,B betűk sorozatával (L=le, J=jobbra, F=fel, B=balra)!

A bemenet első sorában a térkép mérete ($1 \leq N, M \leq 1000$) és a magasságkorlát ($1 \leq H \leq 1000$) van. A második sorban a P és a Q pont sor-, illetve oszlop-indexe van. A következő N sorban soronként M szám található, az egyes pontok tengerszint feletti magassága.

A kimenet első és egyetlen sorába a P és Q közötti leggyorsabb út idejét kell írni! Ha nem lehet eljutni a P pontból a Q pontba, akkor az egyetlen sorba -1-et kell kiírni!

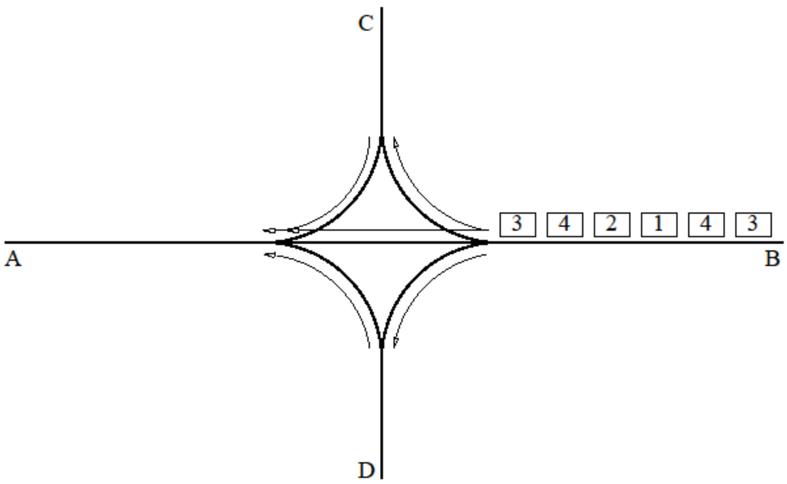
Példa:

Bemenet:	Kimenet:
5 6 4	11
2 4 4 4	
5 5 5 5 5 9	
4 4 4 4 4 9	
5 9 9 9 9 9	
5 9 9 **5** 9 9	
5 5 5 5 5 9	

3. feladat: Vasúti kocsik rendezése (15 pont)

Duplaverem város vasútállomásán sok gondot okoz a vasúti kocsik rendezése. Az állomásról továbbítandó szerelvényeket úgy kell kialakítani, hogy amikor az megérkezik a célállomásra, a szerelvény végéről mindig lekapcsolható legyen az oda továbbított kocsisor. Minden továbbítandó szerelvény négy állomást érint, ezért a rendezés előtt minden kocsit megjelölnek az 1, 2, 3 vagy 4 számokkal. A szerelvény kocsijait át kell rendezni úgy, hogy a szerelvény elején legyenek az 1-essel, aztán a 2-essel, majd a 3-assal, végül a 4-essel megjelöltek. Kezdetben a kocsik az ábrán látható B pályaszakaszon vannak.

A vasúti váltók működése csak a következő műveleteket teszi lehetővé. Az átrendezendő kocsisorból balról az első kocsit át lehet mozgatni vagy az A szakaszba a már ott lévő kocsik mögé, vagy a C vagy D szakaszba a már ott lévő kocsik elé. Továbbá, a C vagy D szakaszon lévő első kocsit át lehet mozgatni az A szakaszon kialakítandó rendezett kocsisor végére.



Készíts programot, amely megállapítja, hogy a beérkező kocsisor rendezhető-e!

A bemenet első sorában a rendezésre váró szerelvények száma van ($1 \leq N \leq 10$). A következő N sor mindegyike egy legfeljebb 1000 kocsit tartalmazó rendezendő szerelvényt ír le, minden sort a 0 szám zárja (ami nem része a bemenetnek). Az utolsó 0 kivételével minden szám értéke 1,2,3 vagy 4.

A kimenetre pontosan N sort kell írni! Az i-edik sorba az IGEN szót kell írni, ha a bemenet i+1-edik sorában szereplő kocsisor rendezhető, egyébként pedig a NEM szót!

Példa:

Bemenet:	Kimenet:
2	IGEN
1 2 3 2 4 0	NEM
2 3 1 4 2 1 0	

4. feladat: Gép kölcsönzés (25 pont)

Mekk Elek vállalkozásának van egy nagy értékű munkagépe, amit más vállalkozók kölcsönözhetnek tőle. Egy vagy több, de összefüggő napokra lehet igényelni, minden napra azonos a bérleti díj. A következő M napra sok megrendelés érkezett. Minden megrendelés két számot tartalmaz, D: ahány napra bérelni akarja a megrendelő, H: az a határidő, ameddig a megrendelőnek el kell végeznie a munkát a géppel. Tehát neki olyan E naptól lehet adni a gépet, amelyre teljesül, hogy E+D-1 ≤ H. Mekk Eleknek az a célja, hogy olyan megrendeléseket teljesítsen, amelyek összességében a lehető legtöbb napra kölcsönzik a gépet.

Készíts programot, amely a megrendelések alapján kiszámítja, hogy Mekk Elek legjobb esetben hány napra tudja bérbe adni a gépet! Továbbá, meg is ad egy megfelelő beosztást.

A bemenet első sorában van a munkanapok száma (1≤M≤100) és a megrendelések száma (1≤N≤1000). A következő N sor mindegyike az igényelt napok számát és a határidőt tartalmazza. A bemenet az igények határideje szerint nem-csökkenően rendezett.

A kimenet első sorába azt a legnagyobb K számot kell írni, amelyre teljesül, hogy összesen K napra bérbe lehet adni a gépet! A második sorban a kielégített igények L száma legyen! A következő L sor egy-egy megrendelés teljesítését tartalmazza: a megrendelés sorszámát és az első nap sorszámát, amelytől a megrendelő használhatja a gépet! A teljesített igények tetszőleges sorrendben megadhatók. Több megoldás esetén bármelyik megadható.

Példa:

Bemenet:	Kimenet:
10 6	9
2 2	4
1 3	6 8
2 3	5 5
3 6	4 2
3 7	2 1
2 9	

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Játékos (18 pont)

Egy N napos játék-versenyen, ahol nem kötelező minden nap játszani, 3 játékos (A,B,C) vesz részt. Nem szerencsések azok a napok, amikor csak 1 játékos vesz részt a versenyen, mert ekkor unatkozhat.

Készíts programot, amely megadja, hogy mikor voltak ilyen helyzetek!

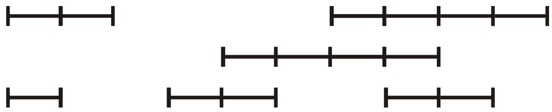
A bemenet első sorában a napok száma van (1≤N≤10 000). A második sorban az A, a harmadikban a B, a negyedikben pedig a C játékos leírása van. Mindhárom sor első száma (1≤M≤N) azt jelenti, hogy a játékos a verseny hány szakaszában vett részt. Ezt M számpár követi: az első tagja a szakasz első napjának sorszáma, a második pedig a szakasz hossza.

A kimenet első sorába azon intervallumokat kell írni, amikor az A, a második sorba azokat, amikor a B, a harmadikba pedig azokat, amikor a C játékos volt egyedül! Minden sor az ilyen intervallumok számával kezdődjön, majd az intervallumok kezdete és vége kövesse, kezdet szerint növekvő sorrendben!

Példa:

```
Bemenet:
10
2 1 2 7 4
1 5 4
3 1 1 4 2 8 2
```

```
Kimenet:
2 2 2 10 10
1 6 6
1 4 4
```



2. feladat: Túra (15 pont)

Ismerjük egy N*M-es térképen a pontok tengerszint feletti magasságát. Két szomszédos pont közötti lépés ideje 1 + a magasságuk különbségének abszolút értéke. Egy pontból nem lehet átlépni a szomszédjába, ha a magasságuk különbsége nagyobb, mint H. Egy P pontból szeretnénk eljutni egy Q pontba.

Készíts programot, amely megadja, hogy P-ből Q-ba minimálisan mennyi idő alatt lehet eljutni és megad egy ilyen utat az L,J,F,B betűk sorozatával (L=le, J=jobbra, F=fel, B=balra)!

A bemenet első sorában a térkép mérete (1≤N,M≤1000) és a magasságkorlát (1≤H≤1000) van. A második sorban a P és a Q pont sor-, illetve oszlop-indexe van. A következő N sorban soronként M szám található, az egyes pontok tengerszint feletti magassága.

A kimenet első sorába a P és Q közötti leggyorsabb út idejét kell írni! A második sorba egy ilyen utat leíró betűsorozat kerüljön! Ha nem lehet eljutni a P pontból a Q pontba, akkor az egyetlen sorba -1-et kell kiírni! Több megoldás esetén bármelyik megadható.

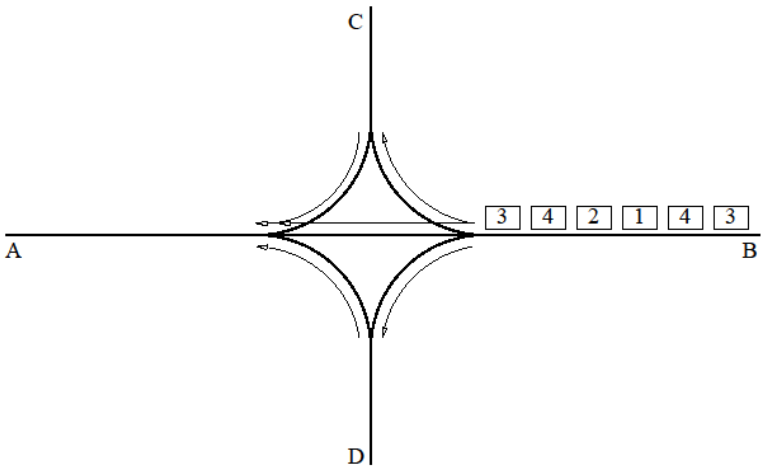
Példa:

```
Bemenet:
5 6 4
2 4 4 4
5 5 5 5 5 9
4 4 4 4 4 9
5 9 9 9 9 9
5 9 9 5 9 9
5 5 5 5 5 9
```

```
Kimenet:
11
BBBLLLJJJF
```

3. feladat: Vasúti kocsik rendezése (14 pont)

Duplaverem város vasútállomásán sok gondot okoz a vasúti kocsik rendezése. Az állomásról továbbítandó szerelvényeket úgy kell kialakítani, hogy amikor az megérkezik a célállomásra, a szerelvény végéről mindig lekapcsolható legyen az oda továbbított kocsisor. Minden továbbítandó szerelvény négy állomást érint, ezért a rendezés előtt minden kocsit megjelölnek az 1, 2, 3 vagy 4 számokkal. A szerelvény kocsijait át kell rendezni úgy, hogy a szerelvény elején legyenek az 1-essel, aztán a 2-essel, majd a 3-assal, végül a 4-essel megjelöltek. Kezdetben a kocsik az ábrán látható B pályaszakaszon vannak.



A vasúti váltók működése csak a következő műveleteket teszi lehetővé. Az átrendezendő kocsisorból balról az első kocsit át lehet mozgatni vagy az A szakaszba a már ott lévő kocsik mögé, vagy a C vagy D szakaszba a már ott lévő kocsik elé. Továbbá a C vagy D szakaszon lévő első kocsit át lehet mozgatni az A szakaszon kialakítandó rendezett kocsisor végére.

Készíts programot, amely megállapítja, hogy a beérkező kocsisor rendezhető-e!

A bemenet első sorában a rendezésre váró szerelvények száma van ($1 \le N \le 20$). A következő N sor mindegyike egy legfeljebb 1000 kocsit tartalmazó rendezendő szerelvényt ír le, minden sort a 0 szám zárja (ami nem része a bemenetnek). Az utolsó 0 kivételével minden szám értéke 1, 2, 3 vagy 4.

A kimenetre pontosan N sort kell írni! Az i-edik sorba az IGEN szót kell írni, ha a bemenet i+1-edik sorában szereplő kocsisor rendezhető, egyébként pedig a NEM szót!

Példa:

Bemenet:	Kimenet:
2	IGEN
1 2 3 2 4 0	NEM
2 3 1 4 2 1 0	

4. feladat: Kötélhúzó verseny (14 pont)

Az osztály kötélhúzó versenyt rendez. A tanulók két csoportot alkotnak, úgy hogy minden tanuló valamelyik csoport tagja lesz, és a két csoport létszáma azonos. Szerencsére az osztály tanulóinak N száma páros szám. Mivel a csoportok tagjainak összsúlya hatással lehet a verseny kimenetelére, ezért úgy kell képezni a két csoportot, hogy a két összsúly eltérése a lehető legkisebb legyen.

Készíts programot, amely kiszámít egy megfelelő csoportbeosztást!

A bemenet első sorában a tanulók száma van ($1 \le N \le 100$, N páros szám). A második sor pontosan N egész számot tartalmaz, az i-edik szám az i-edik tanuló súlya, értéke legfeljebb 100.

A kimenet első sorába a két csoport összsúly-különbségének abszolút értékét kell írni! A második és harmadik sorba kell kiírni a két csoport tagjainak azonosítóit, pontosan N/2 számot! Több megoldás esetén bármelyik megadható.

Példa:

```
Bemenet:                          Kimenet:
8                                 0
44 45 55 56 50 100 80 30          1 4 5 7
                                  2 3 6 8
```

5. feladat: Legnagyobb négyzet (14 pont)

Egy zöldmezős beruházás keretében építendő csarnok elhelyezését kell megtervezni. Feltétel, hogy a csarnok alaprajza négyzet alakú legyen. Megkötés továbbá, hogy a csarnok négy sarokpontját megadott pontok közül kell kiválasztani úgy, hogy a négyzet oldalai párhuzamosak legyenek a koordináta rendszer tengelyeivel. A feltételeknek megfelelő, legnagyobb négyzetet keressük.

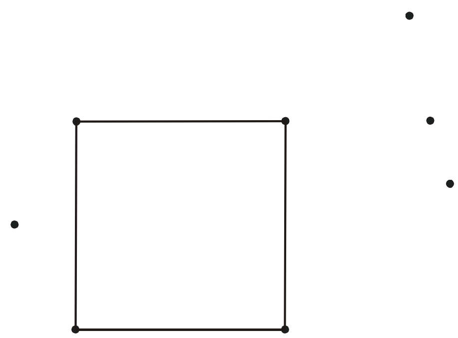
Készíts programot, amely kiszámítja a feltételeknek megfelelő legnagyobb négyzetet!

A bemenet első sorában a lehetséges sarokpontok pontok száma van (`1000≤N≤10 000`). A következő `N` sor mindegyike egy-egy lehetséges sarokpont koordinátái tartalmazza (`1≤X,Y≤ 1 000 000`)).

A kimenet első sorába a lehető legnagyobb négyzet területét kell írni! A 0 érték szerepeljen itt, ha nem lehet négyzetet kialakítani, és ekkor nincs második sor! A második sor a legnagyobb négyzet négy csúcsának sorszámait tartalmazza órajárással ellentétes körüljárás szerint, az bal alsó sarokponttal kezdve! Több megoldás esetén bármelyik megadható.

Példa:

```
Bemenet:                          Kimenet:
8                                 100
1 10                              2 7 3 5
4 5
14 15
22 12
4 15
21 15
14 5
20 20
```



2010. A verseny végeredménye

I. korcsoport

1	Szabó Bence	Fazekas Mihály Gimnázium, Debrecen
2	Palkó András	Vörösmarty Mihály Gimnázium, Szentgotthárd
3	Nagy Vendel	Fazekas Mihály Gimnázium, Debrecen
4	Szabó Áron	Berzsenyi Dániel Evangélikus Gimnázium, Sopron
5	Weisz Ambrus	Fazekas Mihály Gimnázium, Budapest
6	Kocsis Dávid	Ságvári Endre Gimnázium, Szeged
7	Németh Gergely Dániel	Fazekas Mihály Gimnázium, Budapest
8	Géczi Péter	Radnóti Miklós Gimnázium, Szeged
9	Kiss Péter	Fazekas Mihály Gimnázium, Budapest
	Kovács-Deák Máté	Radnóti Miklós Gimnázium, Szeged

II. korcsoport

1	Mezei Balázs	Fazekas Mihály Gimnázium, Budapest
2	Kovács Gábor Ferenc	Árpád Gimnázium, Tatabánya
	Szenczi Zoltán	Veres Péter Gimnázium, Budapest
4	Weisz Gellért	Fazekas Mihály Gimnázium, Budapest
5	Dániel Márton	Árpád Gimnázium, Budapest
	Simon László	Kölcsey Ferenc Főgimnázium, Szatmárnémeti
6	Varnyú József	Fazekas Mihály Gimnázium, Debrecen
7	Erdős Gergely	Batthyány Lajos Gimnázium és Szakközépiskola, Nagykanizsa
8	Kovács Máté	Lovassy László Gimnázium, Veszprém
9	Szabó Attila	Leőwey Klára Gimnázium, Pécs
	Szarka Gábor	Szent István Gimnázium, Budapest

III. korcsoport

1	Hegedűs Tamás	Földes Ferenc Gimnázium, Miskolc
2	Papp Pál András	Szent István Gimnázium, Budapest
3	Hunyady Márton	Bencés Gimnázium, Pannonhalma
4	Mészáros András	Révai Miklós Gimnázium, Győr
5	Palincza Richárd Péter	Berzsenyi Dániel Gimnázium, Budapest
6	Dankovics Attila János	Veres Péter Gimnázium, Budapest
7	Gönczi Tamás	Földes Ferenc Gimnázium, Miskolc
8	Rutai Richárd	Szent István Gimnázium, Budapest
9	Berghammer Tamás	Illyés Gyula Gimnázium és Szakközépiskola, Budaörs
10	Mayer Martin János	Ciszterci Rend Nagy Lajos Gimnázium, Pécs

2011. Első forduló

Ötödik-nyolcadik osztályosok

Számítógép nélküli feladatok

<u>1. feladat</u>: Hibakeresés (25 pont)

Az ország N helységében végeztünk madármegfigyeléseket. Mindegyikben megadtuk, hogy milyen fajú madárból hányat láttunk. A madárfajok száma összesen M. Az alábbi algoritmus megadná azokat a helyeket, ahol mindenféle madár előfordul, ha jó lenne! Jelöld be, mik a hibák benne, majd javítsd is ki!

D(i,j) jelentése: az i-edik helységben a j-edik madárból ennyit láttak.

```
Eljárás(N,M,D):
  Ciklus i=1-től M-ig
    j:=1
    Ciklus amíg j≤M vagy D(i,j)=0
      j:=j+1
    Ciklus vége
    Ha i≥M akkor Ki: i
  Ciklus vége
Eljárás vége.
```

<u>2. feladat</u>: Mit csinál (20 pont)

Az alábbi algoritmus az X vektorban tárolt N szám alapján határozza meg H értékét.

```
Valami(N,X,H):
  i:=1; j:=N
  Ciklus amíg i<j
    Ha X(i)<X(j) akkor i:=i+1 különben j:=j-1
  Ciklus vége
  H:=i
Eljárás vége.
```

A. Mi lesz H értéke, ha N=5 és X=(1,1,1,1,1)?

B. Mi lesz H értéke, ha N=5 és X=(1,2,3,2,1)?

C. Mi lesz H értéke, ha N=5 és X=(2,3,2,4,3)?

C. Mi lesz H értéke, ha N=5 és X=(2,4,2,4,3)?

E. Fogalmazd meg általánosan, mi az eljárás feladata!

<u>3. feladat</u>: Robot (25 pont)

Egy kerekeken guruló robotot az alábbi utasításokkal vezérelhetünk:

- `start(bal,előre)` elindítja a bal kerekeket meghajtó motort, előre
- `start(bal,hátra)` elindítja a bal kerekeket meghajtó motort, hátra
- `stop(bal)` leállítja a bal kerekeket meghajtó motort
- `start(jobb,előre)` elindítja a jobb kerekeket meghajtó motort, előre
- `start(jobb,hátra)` elindítja a jobb kerekeket meghajtó motort, hátra
- `stop(jobb)` leállítja a jobb kerekeket meghajtó motort

A kerekek előremenetből hátramenetbe, illetve fordítva közvetlenül nem kapcsolhatók, kell közben egy leállítás is! Az álló kerék fékezett, azaz a helyéről nem mozdul el.

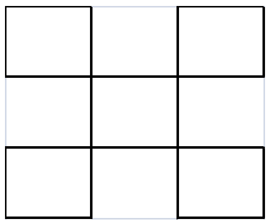
Ha mindkét oldali kerék egy irányban forog, akkor a robot 1 másodperc alatt 1 cm-t tesz meg. Ha csak az egyik forog, akkor 2 másodperc alatt 90 fokot fordul, s ha a két oldali kerék ellenkező irányban forog, akkor 1 másodperc alatt fordul 90 fokot. A program még a következő utasítást használhatja:

- `várj(mp)` `mp` másodperc várakozás

A. Rajzold le a robot útját, amit az alábbi program hatására tesz meg, álló helyzetből indulva!

```
Ciklus i=1-től 4-ig
  start(bal,előre); start(jobb,előre); várj(10)
  stop(jobb); várj(2)
  start(jobb,előre); várj(5)
  stop(bal); start(bal,hátra); várj(1)
  stop(bal); start(bal,előre); várj(5)
  stop(jobb); várj(2); stop(bal)
Ciklus vége
```

B. Készít algoritmust, amelynek alapján a robot az alábbi utat járja be!



Számítógépes feladat – VÁLASZTHATÓ

4. feladat: Balaton (30 pont)

Az elmúlt N napon megmértük a Balatonon a jég vastagságát. Korcsolyázni akkor lehet, ha legalább 2 napon keresztül a vastagság nagyobb volt, mint M centiméter.

Készíts programot, amely

1. megadja, hogy a Balaton hány napon át volt befagyva;
2. megad egy napot, amikor korcsolyázni lehetett;
3. megadja azt a napot, amikor a legvastagabb volt a jég!

Számítógép nélküli feladat – VÁLASZTHATÓ

4. feladat: Hőmérséklet (30 pont)

N napon át minden délben megmértünk a hőmérsékletet. Fagyott az i-edik napon, ha $X(i)<0$, nem fagyott, ha $X(i)\geq 0$. Meg kellene adni

A. az olyan napokat, amelyeken melegebb volt, mint a szomszéd napokon;

B. az olyan nem fagyos napok számát, amikor olyan erős lehűlés kezdődött (azaz előző nap még nem volt csökkenés), hogy a hőmérséklet az előző napi felére esett vissza;

C. a legmelegebb fagyos napot;

D. az első fagyos nap előtti napok átlaghőmérsékletét!

Papíron az alábbi algoritmust kaptuk, egyes helyeken azonban nem látszik az írás. Pótold a hiányos helyeken az algoritmust, hogy az a feladat helyes megoldása legyen!

```
Mérések(N,X):
  dba:=0; dbb:=0
  Ciklus i=2-től N-1-ig
    Ha □□□□□□□□□□□□□□□□ akkor dba:=dba+1; a(dba):=i
    Ha □□□□□□□□□□□□□□□□ akkor dbb:=dbb+1
  Ciklus vége
  c:=0; X(0):=-1000; dbd:=0; sd:=0; volt:=hamis
  Ciklus i=1-től N-ig
    Ha □□□□□□□□□□□□□□□□ akkor c:=i
    Ha X(i)<0 akkor volt:=igaz
    Ha □□□□□□□□□□□□□□□□ akkor dbd:=dbd+1; sd:=sd+X(i)
  Ciklus vége
  Ha □□□□□□□ akkor d:=sd/dbd
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Számok (18 pont)

Az ország N helységében végeztünk madármegfigyeléseket. Mindegyikben megadtuk, hogy milyen fajú madárból hányat láttunk. A madárfajok száma összesen M. Az alábbi algoritmus megadná azokat a madarat, amelyek mindenhol vannak, ahol van valamilyen madár, ha jó lenne! Jelöld be, mik a hibák benne!

D(i,j) jelentése: az i-edik helységben a j-edik madárból ennyit láttak.

```
Eljárás(N,M,D,db,mad):
  db:=0
  Ciklus j=1-től N-ig
    i:=1
    Ciklus amíg i<N vagy nem(van(i) vagy D(i,j)=0)
      i:=i+1
    Ciklus vége
    Ha i>M akkor db:=db+1; mad(i):=j
  Ciklus vége.
Eljárás vége.

van(i):
  k:=1
  Ciklus amíg k≤M és D(k,i)>0
    k:=k+1
  Ciklus vége
  van:=(i≤M)
Függvény vége.
```

2. feladat: Mit csinál (20 pont)

Az alábbi algoritmus az A és a B egész számokból számítja ki a C értékét. A két léptető eljárás a paraméterként kapott értéket egy bittel lépteti balra vagy jobbra.

```
Valami(A,B,C):
  C:=0; D:=A; E:=B
  Ciklus amíg E≠0
    Ha E mod 2=0 akkor Balraléptet(D); Jobbraléptet(E)
              különben C:=C+D; E:=E-1
  Ciklus vége
Eljárás vége.
```

A. Mi lesz C értéke, ha A=1 és B=16?

B. Mi lesz C értéke, ha A=10 és B=8?

C. Mi lesz C értéke, ha A=10 és B=15?

D. Fogalmazd meg általánosan, mi az eljárás feladata!

E. Milyen tulajdonság teljesül az A,B,C,D,E változókra a ciklusfeltétel kiértékelésekor?

<u>3. feladat</u>: Robot (25 pont)

Egy kerekeken guruló robotot az alábbi utasításokkal vezérelhetünk:

- `start(bal,előre)` elindítja a bal kerekeket meghajtó motort, előre
- `start(bal,hátra)` elindítja a bal kerekeket meghajtó motort, hátra
- `stop(bal)` leállítja a bal kerekeket meghajtó motort
- `start(jobb,előre)` elindítja a jobb kerekeket meghajtó motort, előre
- `start(jobb,hátra)` elindítja a jobb kerekeket meghajtó motort, hátra
- `stop(jobb)` leállítja a jobb kerekeket meghajtó motort

A kerekek előremenetből hátramenetbe, illetve fordítva közvetlenül nem kapcsolhatók, kell közben egy leállítás is! Az álló kerék fékezett, azaz a helyéről nem mozdul el.

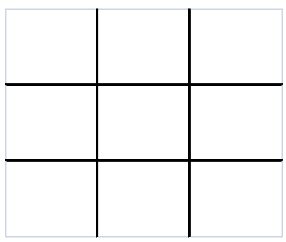
Ha mindkét oldali kerék egy irányban forog, akkor a robot 1 másodperc alatt 1 cm-t tesz meg. Ha csak az egyik forog, akkor 2 másodperc alatt 90 fokot fordul, s ha a két oldali kerék ellenkező irányban forog, akkor 1 másodperc alatt fordul 90 fokot. A program még a következő utasítást használhatja:

- `várj(mp)` `mp` másodperc várakozás

A. Rajzold le a robot útját, amit az alábbi program hatására tesz meg, álló helyzetből indulva!

```
Ciklus i=1-től 4-ig
  start(bal,előre); start(jobb,előre); várj(10)
  stop(jobb); várj(2)
  start(jobb,előre); várj(5)
  Ha i=1 vagy i=3 akkor stop(bal); start(bal,hátra); várj(1)
                        stop(bal); start(bal,előre); várj(5)
                        stop(jobb); várj(2); stop(bal)
  Elágazás vége
Ciklus vége
```

B. Készít algoritmust, amelynek alapján a robot az alábbi utat járja be!



4. feladat: Adatok (28 pont)

Az F(1..6) vektor elemei kezdetben 0 értékűek. Az alábbi 2 eljárást definiáljuk, melyek `a` és `b` paraméterei 1 és 6 közötti egész számok:

```
Egyik(a,b):
  Ciklus amíg F(a)>0
    a:=F(a)
  Ciklus vége
  Ciklus amíg F(b)>0
    b:=F(b)
  Ciklus vége
  Egyik:=(a=b)
Függvény vége.
```

```
Másik-1-változat(a,b):
  Ciklus amíg F(b)>0
    b:=F(b)
  Ciklus vége
  F(b):=a
Eljárás vége.
```

```
Másik-2-változat(a,b):
  Ciklus amíg F(a)>0
    a:=F(a)
  Ciklus vége
  Ciklus amíg F(b)>0
    b:=F(b)
  Ciklus vége
  F(b):=a
Eljárás vége.
```

A. Az alábbi eljáráshívás sorozatra írd le, hogy mi lesz az F vektor értéke az egyes hívások után az 1. változatban!

`Másik(1,3); Másik(2,4); Másik(3,4); Másik(3,5); Másik(6,4)`

B. Az alábbi eljáráshívás sorozatra írd le, hogy mi lesz az F vektor értéke az egyes hívások után a 2. változatban!

`Másik(1,3); Másik(2,4); Másik(3,4); Másik(3,5); Másik(6,4)`

C. Az `egyik` függvényt 15-féle paraméterezéssel hívhatjuk meg, ha feltesszük, hogy `a<b`. Hány esetben lesz igaz a függvény értéke az A részfeladatban szereplő `Másik` eljárás hívásai előtt, illetve egyes hívások után?

5. feladat: Játék (21 pont)

Tekintsük azt a játékot, amelyet N sorból és M oszlopból álló négyzetrácsos táblán játszanak! Egy bábut kell mozgatni a táblán! A bábu kezdetben a tábla bal felső sarkában van, és a jobb alsó sarokba kell eljuttatni az alábbi lépés-szabályt betartva:

- Csak olyan mezőre lehet lépni, ahol még nem lépett a bábu.
- Csak a négy szomszédos mező valamelyikére lehet lépni.
- Egy lépésben csak jobbra, lefelé, vagy felfelé lehet lépni.

Számítsd ki, hogy hányféleképpen lehet eljuttatni a bábut a bal felső sarokból a jobb alsóba!

A. N=2, M=3

B. N=4, M=6

C. N=7, M=3

D. N=7, M=8

E. N=3, M=10

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Számok (14 pont)

Az ország N helységében végeztünk madármegfigyeléseket. Mindegyikben megadtuk, hogy milyen fajú madárból hányat láttunk. A madárfajok száma összesen M. Az alábbi algoritmus megadná azokat a madarakat, amelyek csak valamilyen más madárral együtt fordulnak elő, ha jó lenne! Jelöld be, mik a hibák benne!

D(i,j) jelentése: az i-edik helységben a j-edik madárból ennyit láttak.

```
Eljárás(N,M,D,db,mad):
  db:=0
  Ciklus j=1-től N-ig
    i:=1
    Ha j=i akkor i:=i+1
    Ciklus amíg i≤N és nem jó(i,j)
      i:=i+1
      Ha j=i akkor j:=j+1
    Ciklus vége
    Ha i>M akkor db:=db+1; mad(db):=j
  Ciklus vége
Eljárás vége.
```

```
jó(i,j):
  k:=i
  Ciklus amíg k≤N és (D(i,k)=0 és D(k,j)>0 vagy D(k,j)=0)
    k:=k+1
  Ciklus vége
  jó:=(k>N)
Függvény vége.
```

2. feladat: Mit csinál (17 pont)

Az alábbi algoritmus az X pozitív egész számból számítja ki Y értékét.

```
Valami(X,Y):
  A:=0; B:=1; C:=0
  Ciklus amíg C≤X
    C:=C+B; A:=A+1; B:=B+2
  Ciklus vége
  Y:=A-1
Eljárás vége.
```

A. Mi lesz Y értéke X=16 esetén?

B. Mi lesz Y értéke X=5 esetén?

C. Mi lesz Y értéke X=35 esetén?

D. Fogalmazd meg általánosan, mi az eljárás feladata!

E. Hogyan változnak a ciklusban az A,B,C változók?

3. feladat: Robot (22 pont)

Egy kerekeken guruló robotot az alábbi utasításokkal vezérelhetünk:

- `start(bal,előre)` elindítja a bal kerekeket meghajtó motort, előre
- `start(bal,hátra)` elindítja a bal kerekeket meghajtó motort, hátra
- `stop(bal)` leállítja a bal kerekeket meghajtó motort
- `start(jobb,előre)` elindítja a jobb kerekeket meghajtó motort, előre

- `start(jobb,hátra)` elindítja a jobb kerekeket meghajtó motort, hátra
- `stop(jobb)` leállítja a jobb kerekeket meghajtó motort

A kerekek előremenetből hátramenetbe, illetve fordítva közvetlenül nem kapcsolhatók, kell közben egy leállítás is! Az álló kerék fékezett, azaz a helyéről nem mozdul el.

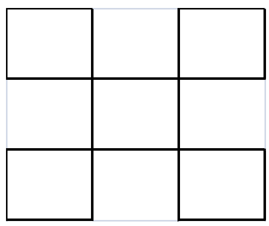
Ha mindkét oldali kerék egy irányban forog, akkor a robot 1 másodperc alatt 1 cm-t tesz meg. Ha csak az egyik forog, akkor 2 másodperc alatt 90 fokot fordul, s ha a két oldali kerék ellenkező irányban forog, akkor 1 másodperc alatt fordul 90 fokot. A program még a következő utasítást használhatja:

- `várj(mp)` `mp` másodperc várakozás

A. Rajzold le a robot útját, amit az alábbi program hatására tesz meg, álló helyzetből indulva, a N=7, H=10 paraméterekkel hívjuk meg!

```
Halad(N,H):
  start(bal,előre); start(jobb,előre); várj(H)
  Ha N>1 akkor stop(bal); várj(2); stop(jobb)
               Halad(N-1,H+10)
               stop(bal); várj(2); stop(jobb)
      különben stop(bal); stop(jobb)
  Elágazás vége
  start(bal,hátra); start(jobb,hátra); várj(H)
Eljárás vége.
```

B. Készít algoritmust, amelynek alapján a robot az alábbi utat járja be!



4. feladat: Adatok (26 pont)

Az F(1..6) vektor elemei kezdetben 0 értékűek. Az alábbi 2 eljárást definiáljuk, melyek a és b paraméterei 1 és 6 közötti egész számok:

```
Egyik-1-változat(a,b):
  Ciklus amíg F(a)>0
    a:=F(a)
  Ciklus vége
  Ciklus amíg F(b)>0
    b:=F(b)
  Ciklus vége
  egyik:=(a=b)
Függvény vége.
```

```
Egyik-2-változat(a,b):
  Belső(a,x); Belső(b,y)
  egyik:=(x=y)
Függvény vége.

Belső(x,y):
  Ha F(x)>0 akkor Belső(F(x),y)
                  F(x):=y
       különben y:=x
Eljárás vége.
```

```
Másik(a,b):
  Ciklus amíg F(a)>0
    a:=F(a)
  Ciklus vége
  Ciklus amíg F(b)>0
    b:=F(b)
  Ciklus vége
  F(b):=a
Eljárás vége.
```

A. Az alábbi eljáráshívás sorozatra írd le, hogy mi lesz az F vektor értéke az egyes hívások után!

`Másik(1,3); Másik(2,4); Másik(3,4); Másik(3,5); Másik(6,4)`

B. Az `egyik` függvényt 15-féle paraméterezéssel hívhatjuk meg, ha feltesszük, hogy `a<b`. Hány esetben lesz igaz a függvény első változatának értéke a `Másik` eljárás hívásai előtt, illetve az A részfeladatban szereplő egyes hívások után?

C. Mi lesz az F vektor értéke az egyes hívások után, ha az `Egyik` függvény második változatát hívjuk?

```
Másik(1,3); Másik(2,4); Másik(3,4); t:=Egyik(3,4);
Másik(5,3); t:=Egyik(2,3); Másik(6,4)
```

<u>5. feladat</u>: Fűrészmalom (21 pont)

A folyó mentén kitermelt fát N helyen gyűjtik össze és szállítják a folyón lefelé úsztatva az első gyűjtőhelyre, ahol fűrészmalomban végzik a feldolgozást. A vállalat elhatározta, hogy további fűrészmalmot állít üzembe néhány gyűjtőhelyen. Minden gyűjtőhelyről a fát a folyón lefelé haladva az első fűrészmalomba fogják szállítani. A szállítási költség a megtett távolság és a tömeg szorzata. Ismerjük az egyes gyűjtőhelyek elhelyezkedését (az elsőtől vett távolságot km-ben) és azt, hogy mennyi fa keletkezik évente az egyes gyűjtőhelyen. Kiszámítandó, hogy hova kell telepíteni az új fűrészmalmokat, hogy a szállítási összköltség a lehető legkisebb legyen!

A táblázat első sora a gyűjtőhely sorszámát, a második sor az 1. gyűjtőhelytől vett távolságot, a harmadik sor a gyűjtőhelyen keletkező fa tömegét tartalmazza.

1.	2.	3.	4.	5.	6.	7.	8	9	10	11	12	13	14.
0	2	3	6	7	20	22	34	35	44	57	66	88	100
1	2	3	44	5	6	33	18	9	10	11	2	13	44

Hová kell telepíteni?

A. További 1 fűrészmalmot

B. További 2 fűrészmalmot

C. További 3 fűrészmalmot

D. További 4 fűrészmalmot

2011. Második forduló

Ötödik-nyolcadik osztályosok

1. feladat: Sípálya (25 pont)

Egy hójelentés N sípályán mért hóréteget tartalmazza.

Készíts programot, amely beolvassa a sípályák számát ($1 \le N \le 20$) és az egyes pályákon a hóréteg vastagságát ($0 \le V(i) \le 100$), majd

A. megadja, hogy melyik sípályán a legnagyobb a hóréteg;

B. megad egy sípályát, ahol a hóréteg legalább 100 cm vastag;

C. megadja azokat a sípályákat, ahol nem lehet síelni (azaz a hóréteg vastagsága 0)!

Példa:

Bemenet:	Kimenet:
Pályák száma: 7	Legnagyobb: 3
1. pálya: 0	Van 100 cm hó: 2
2. pálya: 150	Nem lehet síelni 3 pályán: 1 4 6
3. pálya: 200	
4. pálya: 0	
5. pálya: 57	
6. pálya: 0	
7. pálya: 30	

2. feladat: Pénz (30 pont)

Bergengóciában N-féle pénzérmét használnak: P(1), P(2), …, P(N) forintost.

Írj programot, amely megadja, hogy mely 1 és M forint közötti összegek fizethetők ki egyetlen pénzérmével, melyek legfeljebb 2 pénzérmével és melyek legfeljebb 3 pénzérmével!

A program olvassa be a pénzérmék számát ($1 \le N \le 10$), és a kifizetendő összeget ($1 \le M \le 1000$)! Ezután olvassa be a pénzérmék értékét ($1 \le P_i \le M$)!

A program írja ki a legfeljebb 1, 2, majd a 3 pénzérmével kifizethető összegeket!

Példa:

Bemenet:	Kimenet:
Érmék száma: 2	1 érmével: 1 5
Maximális összeg: 100	2 érmével: 1 2 5 6 10
1. érme: 1	3 érmével: 1 2 3 5 6 7 10 11 15
2. érme: 5	

3. feladat: Nyelv (20 pont)

Egy programozási nyelven az elágazások az `IF` szóval kezdődnek és a `FI` szóval végződnek. Minden program legalább egy, legfeljebb 100 szóból áll

Készíts programot, amely beolvas szavanként egy szöveget, majd megadja, hogy az elágazások egymásba ágyazása helyes-e! Ha nem helyes, akkor megadja az első hiba okát is. (Pl. hibás egymásba ágyazás az alábbi: … `IF` … `FI` … `FI` … `IF` …)

Megjegyzés: a többi szó bármi lehet, ellenőrzésükkel nem kell foglalkozni.

```
Bemenet:                  Kimenet:
Szavak száma: 7           Hibás: FI utasítás hiányzik
1. szó: IF
2. szó: ALMA
3. szó: IF
4. szó: FI
5. szó: FI
6. szó: IF
7. szó: BARACK
```

Kilencedik-tizedik osztályosok

1. feladat: Falu (18 pont)

Ismerjük egy megye települései (falvak, városok) közötti utak hosszát. Zsákfalunak nevezzük azt a falut, ahova csak egyetlen út vezet (és onnan tovább már nem lehet menni, csak visszafelé). A településeket sorszámmal azonosítjuk.

Készíts programot, amely megadja:

A. a zsákfalvak számát;

B. azt a települést, ahova a legtöbb út vezet szomszédos településről;

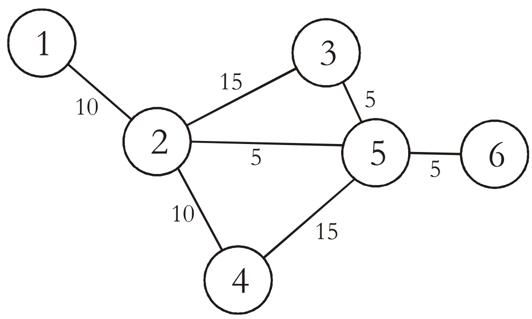
C. az egymáshoz legközelebbi 2, nem szomszédos települést (ha több ilyen van, akkor bármelyik megadható)!

A bemenet első sorában a települések (2≤N≤1000) és az utak száma van (1≤M≤100 000). A következő M sor mindegyikében egy-egy út két végpontjának sorszáma és a köztük levő út hossza van.

A kimenet első sorába a zsákfalvak számát; a második sorba a legtöbb utas település sorszámát (ha több van, bármelyik megadható), a harmadikba pedig a két legközelebbi település sorszámát (ha több egyforma távolság is van, bármelyik településpár megadható) kell írni!

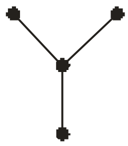
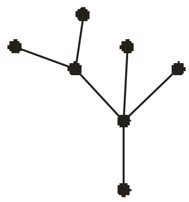
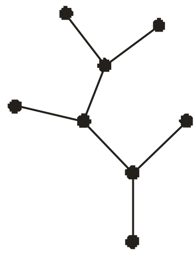
Példa:

```
Bemenet:          Kimenet:
6 7               2
1 2 10            2
2 3 15            3 6
2 4 10
2 5 5
3 5 5
4 5 15
5 6 5
```



<u>2. feladat</u>: Fa (18 pont)

Minden fát leírhatunk egy karaktersorozattal. Ebben a leírásban X betűk és zárójelek fognak szerepelni. Az X ágat jelent, az ágak végi elágazásokat pedig zárójelbe tesszük.

			
X	X(X)(X)	X(X(X)(X))(X)(X)	X(X(X)(X(X)(X))(X)

Írj programot, amely megadja:

A. a fa magasságát (a földtől milyen messze van a legmesszebb levő ágvég);

B. a fa elágazásai számát (a törzs nem számít elágazásnak);

C. egy helyen a legnagyobb elágazásszámot!

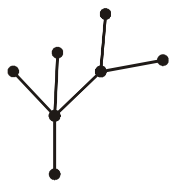
A bemenet egyetlen sorában a fát leíró szöveg van (hossza legfeljebb 10000 karakter).

A kimenet első sorába a fa magasságát, a második sorába a fa elágazásai számát, a harmadik sorába pedig a legnagyobb elágazásszámot kell kiírni!

<u>Példa</u>:

Bemenet:	Kimenet:

`X(X)(X)(X(X)(X))`	`3`
	`5`
	`3`



<u>3. feladat</u>: Pakolás (18 pont)

Egy raktárban egyetlen hosszú sorban ládák vannak. Minden láda kocka alakú, de méretük különböző lehet. A ládák egymásra rakásával akarnak helyet felszabadítani. A biztonsági előírás szerint több ládát is lehet egymásra rakni, de minden ládát csak nálánál nagyobbra lehet helyezni. Továbbá, az i-edik helyen lévő ládát csak akkor lehet ráraknia j-edik helyen lévő torony tetejére, ha az i-edik és j-edik helyek között már nincs láda (j lehet akár kisebb, akár nagyobb, mint i). Minden ládát legfeljebb egyszer lehet mozgatni.

Készíts programot, amely kiszámítja, hogy legkevesebb hány toronyba lehet a ládákat összepakolni!

A bemenet első sorában a ládák száma van (2≤N≤30 000). A második sor a ládák méreteit tartalmazza (1 és 30 000 közötti értékek).

A kimenet első és egyetlen sora azt a legkisebb M számot tartalmazza, hogy a bementben megadott ládasor összepakolható M számú toronyba!

<u>Példa</u>:

Bemenet:	Kimenet:
`10` `1 2 4 6 7 5 3 2 5 3`	`2`

4. feladat: Játék (21 pont)

Tekintsük azt az egyszemélyes játékot, amelyet N sorból és M oszlopból álló négyzetrácsos táblán játszanak. Minden mező vagy üres, vagy csapda. Egy bábut kell mozgatni a táblán. A bábu kezdetben a tábla bal felső sarkában van, és a jobb alsó sarokba kell eljuttatni az alábbi lépés-szabályt betartva:

- Csak olyan mezőre lehet lépni, ahova még nem lépett a bábu.
- Csapda mezőre nem lehet lépni.
- Csak a négy szomszédos mező valamelyikére lehet lépni.
- Egy lépésben csak jobbra, vagy lefelé lehet lépni.

Készíts programot, amely kiszámítja, hogy hányféle képen lehet eljuttatni a bábut a bal felső sarokból a jobb alsóba!

A bemenet első sorában a sorok és oszlopok M száma van ($1 \leq N \leq 10$, $1 \leq M \leq 20$). A további N sor mindegyike M mező leírását tartalmazza. Minden szám vagy 0, vagy 1. A sorban az i-edik szám 1, akkor a megfelelő mező csapda, egyébként a mező üres.

A kimenet egyetlen sora azt tartalmazza, hogy hányféleképpen lehet eljuttatni a bábut a bal felső sarokból a jobb alsóba!

Példa:

Bemenet:	Kimenet:
5 6	7
0 0 0 0 0 0	
0 1 0 0 1 0	
0 0 1 0 0 0	
1 0 1 0 0 0	
0 0 0 0 1 0	

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Falvak (15 pont)

Ismerjük egy megye települései (falvak, városok) közötti utak hosszát. Zsákfalunak nevezzük azt a falut, ahova csak egyetlen út vezet (és onnan tovább már nem lehet menni, csak visszafelé). A településeket sorszámmal azonosítjuk.

Készíts programot, amely megadja:

A. a leghosszabb zsákfalvakhoz vezető út hosszát, ahol nem lehet más irányba letérni (csak visszafordulni);

B. azokat a településeket, ahova a legtöbb út vezet (ha több ilyen van, akkor mindegyiket);

C. azt a települést, amelyiktől a legközelebbi szomszédja a lehető legmesszebb van (ha több ilyen van, akkor a legkisebb sorszámút)!

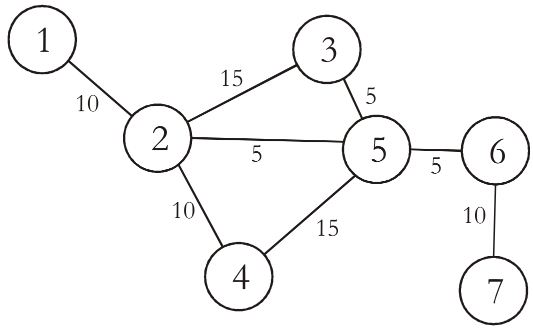
A bemenet első sorában a települések száma ($2 \leq N \leq 1000$) és az utak száma ($1 \leq M \leq 100\ 000$) van. A következő M sor mindegyikében egy-egy út két végpontjának sorszáma, valamint a köztük levő út hossza található.

A kimenet első sorába a leghosszabb zsákfalvakhoz vezető út hosszát; a második sorba a legtöbb utas települések sorszámát (ha több van, akkor sorszám szerint növekvő sorrendben), a harmadikba pedig a település sorszámát kell írni!

Példa:

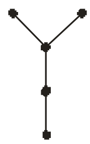
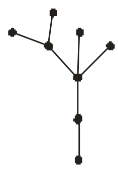
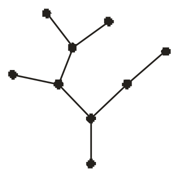
Bemenet:
```
7 8
1 2 10
2 3 15
2 4 10
2 5 5
3 5 5
4 5 15
5 6 5
6 7 10
```

Kimenet:
```
15
2 5
1
```



2. feladat: Fa (15 pont)

Minden fát leírhatunk egy karaktersorozattal. Ebben a leírásban X betűk és zárójelek fognak szerepelni. Az X egységnyi hosszúságú ágat jelent, az ágak végi elágazásokat pedig zárójelbe tesszük.

			
XXX	XX(X)(X)	XX(X(X)(X))(X)(X)	X(X(X)(X(X)(X))(XX)

Írj programot, amely megadja:

A. a fa magasságát (a földtől milyen messze van a legmesszebb levő ágvég);

B. a fa elágazásai számát (a törzs nem számít elágazásnak);

C. a leghosszabb, elágazás nélküli ágszakasz hosszát (a törzs is ágnak számít, az elágazások nem tartoznak az ágszakaszhoz)!

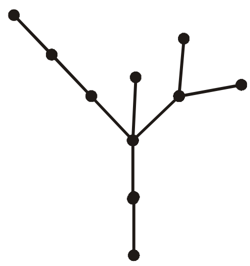
A bemenet egyetlen sorában a fát leíró szöveg van (hossza legfeljebb 10000 karakter).

A kimenet első sorába a fa magasságát, a második sorába a fa elágazásai számát, a harmadik sorába pedig a leghosszabb ág hosszát kell kiírni!

Példa:

Bemenet:
```
XX(XXX)(X)(X(X)(X))
```

Kimenet:
```
5
5
3
```



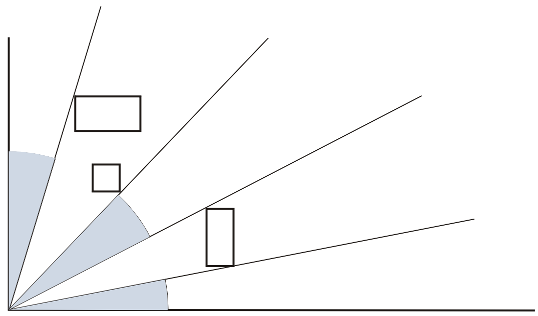
3. feladat: Takar (15 pont)

A síkon, az első síknegyedben elhelyeztünk téglalap alakú épületeket. Az origóból nézve ezek adott szögtartományt takarnak. A szögeket az x-tengelytől számítva, fokban adjuk meg.

Készíts programot, amely megadja, hogy mekkora azon szögtartományok összege, ahol lehet épületet látni!

A bemenet első sorában az épületek száma van ($1 \leq N \leq 1000$). A következő N sorban egy-egy épület által takart szögtartomány kezdete és vége található ($0 \leq K_i < V_i \leq 90$).

A kimenet egyetlen sort kell írni, a takart szögtartomány összegét!



Példa:

Bemenet:
```
3
15 30
55 80
50 60
```

Kimenet:
```
45
```

4. feladat: Pakolás (15 pont)

Egy raktárban egyetlen hosszú sorban ládák vannak. Minden láda kocka alakú, de méretük különböző lehet. A ládák egymásra rakásával akarnak helyet felszabadítani. A biztonsági előírás szerint több ládát is lehet egymásra rakni, de minden ládát csak nálánál nagyobbra lehet helyezni. Továbbá, az i-edik helyen lévő ládát csak akkor lehet rárakni a j-edik helyen lévő torony tetejére, ha az i-edik és j-edik helyek között már nincs láda (j lehet akár kisebb, akár nagyobb, mint i). Minden ládát legfeljebb egyszer lehet mozgatni!

Készíts programot, amely kiszámítja, hogy legkevesebb hány toronyba lehet a ládákat összepakolni!

A bemenet első sorában a ládák száma van (2≤N≤30 000). A második sor a ládák méreteit tartalmazza (1 és 30 000 közötti értékek).

A kimenet egyetlen sorába azt a legkisebb M számot kell írni, ahány toronyba a bemenetben megadott ládasor összepakolható!

Példa:

Bemenet:
```
10
1 2 4 6 7 5 3 2 5 3
```

Kimenet:
```
2
```

5. feladat: Játék (15 pont)

Tekintsük azt az egyszemélyes játékot, amelyet N sorból és M oszlopból álló négyzetrácsos táblán játszanak! A táblán minden mező vagy csapda, vagy valahány gyöngyöt tartalmaz. Egy bábut kell mozgatni a táblán! A bábu kezdetben a tábla bal felső sarkában van, és a jobb alsó sarokba kell eljuttatni az alábbi lépés-szabályt betartva:

- Csak olyan mezőre lehet lépni, ahova még nem lépett a bábu.
- Csapda mezőre nem lehet lépni.
- Csak a négy szomszédos mező valamelyikére lehet lépni.
- Egy lépésben csak jobbra, lefelé, vagy felfelé lehet lépni.
- Minden olyan mezőn lévő gyöngy a játékosé lesz, amely mezőre lép.

A játék célja, hogy a játékos a lehető legtöbb gyöngyöt megszerezze.

Készíts programot, amely kiszámítja a megszerezhető legtöbb gyöngyök számát, és meg is ad egy olyan lépéssorozatot, amely ezt eredményezi!

A bemenet első sorában a sorok és oszlopok száma van (1≤N,M≤100). A további N sor mindegyike M mező leírását tartalmazza. A sorban az i-edik szám a sorban az i-edik mezőn lévő gyöngyök száma, ha a szám nemnegatív. Ha a szám -1, akkor az a mező csapda. Minden szám értéke legfeljebb 2000. A bal felső és a jobb alsó mező biztosan nem csapda, és a kiindulási bal felső mezőn lévő gyöngyök száma beleszámít az összpontszámba.

A kimenet első sora a megszerezhető legtöbb gyöngy számát tartalmazza! Ha nem lehet eljutni a jobb alsó sarokba, akkor a -1 számot kell kiírni, és nem kell második sort írni, egyébként a második sor olyan „J”, „L”, „F” betűkből álló lépéssorozatot tartalmazzon, amely a legtöbb gyöngyöt eredményez! A második sor szóközöket nem tartalmazhat, és sorvége jellel záruljon! A betűk jelentése: J: jobbra, L: lefelé, F: felfelé.

Példa:

```
Bemenet:
5 6
0 0 0 0 -1 0
0 1 0 0 2 0
0 0 -1 1 0 0
3 0 1 0 0 0
0 0 0 0 4 0
```

```
Kimenet:
11
LLLLJFFFFJJLLLLJFFFJLLL
```

2011. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Időjárás (25 pont)

Az időjárás előrejelzésben ismerjük előre N nap (2≤N≤100) várható minimális és maximális hőmérsékletét. Készíts programot, amely beolvassa N értékét és a 2*N db hőmérsékletet, majd megadja:

A. azt a K napos időtartamot (ha van), amelyben az előrejelzés szerint folyamatosan fagy lesz;

B. azt a két szomszédos napot, ahol a legnagyobbat változik a hőmérséklet;

C. azokat a napokat, ahol a napi minimális hőmérséklet a napi átlaghőmérsékletek átlaga fölötti!

Példa:

Bemenet:	Kimenet:
Napok száma?5	Folyamatos fagy: 3 4
Fagy hány napon keresztül?2	Legnagyobb változás: 4 5
1. nap minimuma, maximuma: -9 -2 2. nap minimuma, maximuma: -1 4 3. nap minimuma, maximuma: -5 -4 4. nap minimuma, maximuma: -6,-1 5. nap minimuma, maximuma: 5 8	Átlag fölöttiek: 2 5

2. feladat: Webcím (25 pont)

Wikipédián található az alábbi leírás a webcímekről (most csak azt engedjük meg, ami ebben a leírásban szerepel):

Egy tipikus, egyszerű webcím így néz ki:

http://hu.wikipedia.org:80/wiki

Ennek részei:

- *A http (vagy https) a használandó protokoll. A protokoll neve után kettőspont (:) írandó.*
- *A hu.wikipedia.org a célgép tartományneve. Ez elé két perjel (//) írandó.*
- *A 80 a célgép azon hálózati portszáma, amin kérésünket várja; ez elé kettőspont (:) írandó. Ezt a részt gyakran teljesen elhagyhatjuk, például esetünkben a http protokoll alapértelmezett portszáma a 80.*
- *A /wiki a kért elérési út a célgépen. Ez a rész mindig a perjellel (/) kezdődik.*

A legtöbb böngésző nem is igényli, hogy a „http://” részt begépeljük egy weblap eléréséhez, hiszen az esetek döntő többségében úgyis ezt használjuk. Egyszerűen begépelhetjük a lap címét, például: „hu.wikipedia.org/wiki/Bit”. A főlap megtekintéséhez általában elég a tartomány nevét beírni, például „hu.wikipedia.org”.

A webcím a példákban szereplőtől eltérő jeleket (pl. szóközt, relációkat, …) nem tartalmazhat.

A webcímek egyéb részeket is tartalmazhatnak, http esetében például az elérési út után, egy kérdőjel (?) mögé helyezve keresési kérdés szerepelhet, ami egy get metódusú HTML űrlapból származik.

Példa:

http://hu.wikipedia.org/w/wiki.phtml?title=Bit&action=history

Készíts programot, amely egy beolvasott webcímet (legfeljebb 100 karakteres) a fenti szempontok szerint ellenőriz! Ha valamely szempont szerint hibás, akkor megadja, hogy mi a hiba a szövegben.

Példa:

```
Bemenet: http://www.njszt.hu\tehetseg
Kimenet: Hibás karakter: \
```

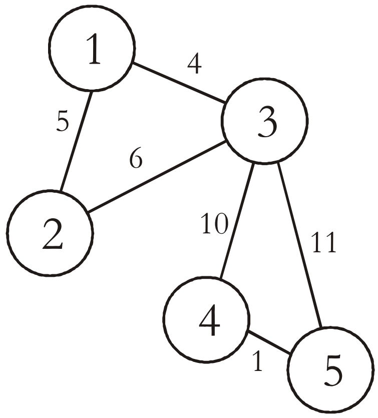
3. feladat: Elszigetelt falu (25 pont)

Egy megyében N falu van (1≤N≤100). A falvakat M út köti össze (3≤M≤1000), ismerjük minden út hosszát. A legelszigeteltebb falunak azt nevezzük, amelytől a legközelebbi szomszédja a lehető legtávolabb van.

Készíts programot, amely megadja a legelszigeteltebb falut! Ha több megoldás van, akkor bármelyik kiírható.

Példa:

```
Bemenet:
Falvak száma?5
Utak száma?6
1. út kezdete, vége, hossza?1 2 5
2. út kezdete, vége, hossza?2 3 6
3. út kezdete, vége, hossza?3 1 4
4. út kezdete, vége, hossza?3 4 10
5. út kezdete, vége, hossza?4 5 1
6. út kezdete, vége, hossza?3 5 11
Kimenet:
A legelszigeteltebb: 2
```



Kilencedik-tizedik osztályosok

1. feladat: Fénykép (19 pont)

Egy rendezvényre sok vendéget hívtak meg. Minden vendég előre megadta, hogy mikor érkezik, és mikor távozik. A rendezők fényképen akarják megörökíteni a résztvevőket. A munkára kiválasztott fényképész úgy dolgozik, hogy egy menetben lefényképezi mindazokat, akik a menet F kezdete és F+K vége közötti időintervallumban jelen voltak a rendezvényen. Pontosabban lefényképez minden olyan vendéget, akinek E érkezési és T távozási idejére teljesül, hogy E<F+K, és F≤T. A fényképészt a menetek száma szerint kell fizetni, tehát az a cél, hogy a lehető legkevesebb menet legyen, de mindenki rajta legyen legalább egy fényképen.

Készíts programot, amely megadja, hogy legkevesebb hány menetre van szükség, hogy mindenki rajta legyen legalább egy fényképen, és meg is adja, hogy mikor kezdődjenek a menetek!

A bemenet első sorában a vendégek száma (`1≤N≤100 000`), és a menetek hosszát megadó szám (`2≤K≤1000`) van. A további N sor mindegyikében egy-egy vendég érkezési és távozási ideje található (`1≤E<T<20 000`).

A kimenet első sorába a minimálisan szükséges menetek M számát kell írni! A második sor pontosan M egész számot tartalmazzon, az egyes menetek kezdő időpontját! Több megoldás esetén bármelyik megadható.

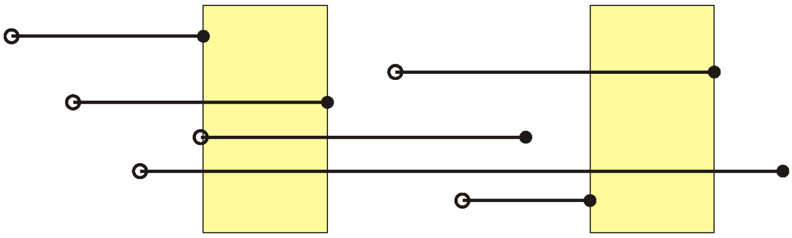
Példa:

Bemenet:

```
6 2
1 4
7 12
2 6
4 9
3 13
8 10
```

Kimenet:

```
2
4 10
```



2. feladat: Suduko (18 pont)

A Suduko játék 4×4-es változatában egy 4×4-es táblázatot kell kitölteni úgy, hogy minden sorban, minden oszlopban és mind a négy sarok 2×2-es részben az 1,2,3,4 számok mindegyike előforduljon!

Készíts programot, amely megadott táblázatot kitölt, hogy az suduko táblázat legyen!

A bemenet négy sort tartalmaz, minden sorban négy egész szám van. A számok értéke 0,1,2,3,4 lehet. A 0 azt jelenti, hogy a táblázat azon elemét meg kell adni a megoldásban. Feltételezhető, hogy minden bemenetre van megoldás.

A kimenetre a bemenetben megadott nem teljesen kitöltött táblázat egy szabályos kitöltését kell írni! Négy sort kell kiírni, minden sorban négy egész számot! Több megoldás esetén bármelyik megadható.

Példa:

Bemenet:

```
0 3 0 1
0 1 0 3
3 0 1 0
0 2 0 4
```

Kimenet:

2 3 **4** 1
4 1 **2** 3
3 **4** 1 **2**
1 2 **3** 4

3. feladat: Jelentés (18 pont)

Egy vállalat hierarchikus felépítésű, tehát az igazgató kivételével (akinek nincs főnöke) minden dolgozónak pontosan egy közvetlen főnöke van, továbbá az igazgató mindenkinek a főnöke (közvetlenül, vagy közvetve). Ha egy dolgozó jelentést akar küldeni az igazgatónak, akkor be kell tartania a szolgálati utat, ami azt jelenti, hogy a jelentés egy lépésben, azaz egy nap alatt a dolgozó közvetlen főnökéhez kerül, aki azt továbbítja az közvetlen főnökének, és így tovább, amíg az üzenet meg nem érkezik az igazgatóhoz.

Készíts programot, amely kiszámítja, hogy legrosszabb esetben hány nap kell ahhoz, hogy az igazgató megkapja dolgozója jelentését! Továbbá, meg is ad egy ilyen dolgozót!

A bemenet első sorában a dolgozók száma van ($1 \le N \le 1000$). Az igazgató azonosítója az 1. A következő N sor a vállalati hierarchiát írja le. Az állomány i+1-edik sorában azok a dolgozók vannak felsorolva (a felsorolást a 0 szám zárja), akik az i-edik dolgozó közvetlen beosztottjai, tehát akiknek közvetlen főnöke i. Ha nincs egyetlen közvetlen beosztottja sem, akkor a sor csak a 0-t tartalmazza.

A kimenet első sorába azt a legnagyobb M számot kell írni, ahány nap szükséges ahhoz, hogy egy jelentés eljusson az igazgatóhoz! A második sor egy olyan dolgozó azonosítóját tartalmazza, akitől a jelentés M nap alatt jut az igazgatóhoz! Ha több ilyen van, akkor a legkisebbet kell kiírni!

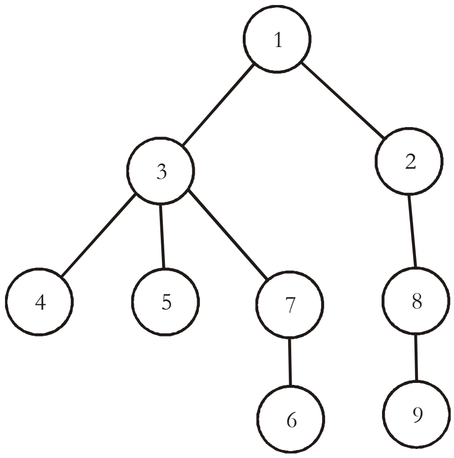
Példa:

Bemenet:

```
9
3 2 0
8 0
4 5 7 0
0
0
0
6 0
9 0
0
```

Kimenet:

```
3
6
```



4. feladat: Csapat (20 pont)

Egy kiránduláson részvevő tanulókból két csapatot kell képezni! A két csapatot úgy kell képezni, hogy ha X és Y nem szeretik egymást, akkor különböző csapatba kerüljenek! Mindkét csapat legalább egy tanulót tartalmazzon!

Készíts programot, amely kiszámít egy, a feltételeknek megfelelő csapat beosztást!

A bemenet első sorában a tanulók száma (`1≤N≤200`) és azon párok száma van (`1≤M≤20 000`), akik nem szeretik egymást. A következő M sor mindegyike két tanuló sorszámát tartalmazza (`1≤X,Y≤N`), ami azt jelenti, hogy X és Y nem szeretik egymást.

A kimenetre két sort kell kiírni, soronként a két csapat tagjait! Ha nincs megoldása a feladatnak, akkor az első és egyetlen sorba a -1 számot kell kiírni! A sorban a számok tetszőleges sorrendben kiírhatók. Több megoldás esetén bármelyik megadható.

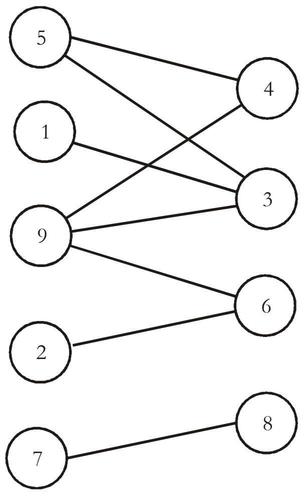
Példa:

Bemenet:

```
9 8
1 3
3 9
9 4
5 4
2 6
7 8
3 5
9 6
```

Kimenet:

```
1 2 5 7 9
3 4 6 8
```



Tizenegyedik-tizenharmadik osztályosok

1. feladat: Párok (15 pont)

Egy rendezvényre sok vendéget hívtak meg. Minden vendég előre megadta, hogy mikor érkezik, és mikor távozik. A rendezők fényképeken akarják megörökíteni a résztvevőket. A rendezőknek két betartandó kikötése van:

1. Minden képen pontosan két vendég legyen rajta.

2. Minden vendég legfeljebb egy képen szerepelhet.

Természetesen két vendég csak akkor szerepelhet azonos képen, ha van olyan F időpont, amikor mindketten jelen vannak. Egy vendég akkor és csak akkor van jelen az F időpontban, ha az E érkezési és T távozási idejére teljesül, hogy $E \le F$, és $F < T$.

Készíts programot, amely kiszámítja, hogy legjobb esetben hány fénykép készülhet, és megadja, hogy mely párok szerepeljenek egy képen!

A bemenet első sorában a vendégek száma van ($1 \le N \le 30\,000$). A további N sor mindegyikében egy-egy vendég érkezési és távozási ideje található ($1 \le E < T < 20\,000$).

A kimenet első sorába a lehetséges legtöbb készíthető fényképek M számát kell írni! A további M sor mindegyikébe azon két vendég sorszámát kell írni (tetszőleges sorrendben), akik egy képen szerepelnek! Több megoldás esetén bármelyik megadható.

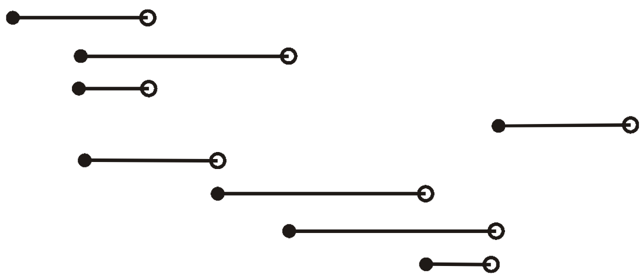
Példa:

Bemenet:

```
8
1 3
2 5
2 3
8 10
2 4
4 7
5 8
7 8
```

Kimenet:

```
3
3 1
5 2
6 7
```



2. feladat: Sudoku (15 pont)

A Sudoku játék 4×4-es változatában egy 4×4-es táblázatot kell kitölteni úgy, hogy minden sorban, minden oszlopban és mind a négy sarok 2×2-es részben az 1,2,3,4 számok mindegyike előforduljon!

Készíts programot, amely adott, nem teljesen kitöltött táblázatra kiszámítja, hogy hány féleképpen lehet kitölteni úgy, hogy szabályos sudoku táblázat legyen!

A bemenet négy sort tartalmaz, minden sorban négy egész szám van. A számok értéke 0,1,2,3,4 lehet. A 0 azt jelenti, hogy a táblázat azon elemét meg kell adni a megoldásban.

A kimenet első és egyetlen sorába azt kell írni, ahányféleképpen kitölthető a bemeneti táblázat szabályos sudoku táblázattá!

Példa:

Bemenet:

```
0 3 0 1
0 1 0 3
3 0 1 0
0 2 0 4
```

Kimenet:

```
2
```

Segítségként a 2 megoldás:

```
2 3 4 1
4 1 2 3
3 4 1 2
1 2 3 4
```

```
4 3 2 1
2 1 4 3
3 4 1 2
1 2 3 4
```

3. feladat: Kemence (15 pont)

Cserépkorsók kiégetésére szakosodott vállalkozó egy kemencét üzemeltet. Az égetésre érkező korsókat az érkezés sorrendjében kell a kemencében kiégetni. Egy menetben legfeljebb K korsó rakható a kemencébe. Minden korsóra adott a minimális és maximális égetési idő percben kifejezve. Továbbá, minden korsóra adott egy H határidő, ami azt jelenti, hogy a munka megkezdésétől számítva, a H időpontig el kell készülnie a kiégetésének. Figyelembe kell venni, hogy egy menet előkészítése 1 percet vesz igénybe!

Készíts programot, amely kiszámítja, hogy a követelmények betartásával legkevesebb mennyi idő alatt lehet kiégetni az összes korsót és meg is ad egy helyes beosztást!

A bemenet első sorában a korsók száma (`1≤N≤10 000`), és a kemence kapacitása (`1≤K≤100`) van. A következő N sor mindegyike a minimális és a maximális égetési időt tartalmazza (percben megadva, `1≤min≤max≤1000`), valamint a határidő értékét percben megadva.

A kimenet első sorába az összes korsó kiégetéséhez szükséges legkisebb időt és az égetési menetek M számát kell írni! A következő M sor mindegyike az u v számpárt tartalmazza, ami azt jelenti, hogy ebben a menetben az `u,u+1,…,v` sorszámú korsók kerülnek a kemencébe (`1≤u≤v≤N`, `v≤u+K-1`)! Több megoldás esetén bármelyik megadható. A feladat minden tesztesetre megoldható.

Példa:

Bemenet:	Kimenet:
4 3	9 3
1 2 4	1 2
2 3 3	3 3
3 4 8	4 4
1 2 9	

4. feladat: Képzés (15 pont)

Egy vállalat fel akarja készíteni a dolgozóit egy új szoftver használatára. Arra nincs lehetőség, hogy minden dolgozó részt vegyen kiképzésen. Ezért az igazgató elhatározta, hogy a lehető legkevesebb dolgozó vegyen részt kiképzésen, de teljesüljön, hogy ha egy dolgozó nem vett részt a kiképzésen, akkor a közvetlen főnöke biztosan részt vett. A vállalat hierarchikus felépítésű, tehát az igazgató kivételével (akinek nincs főnöke) minden dolgozónak pontosan egy közvetlen főnöke van, továbbá az igazgató mindenkinek a főnöke (közvetlenül, vagy közvetve).

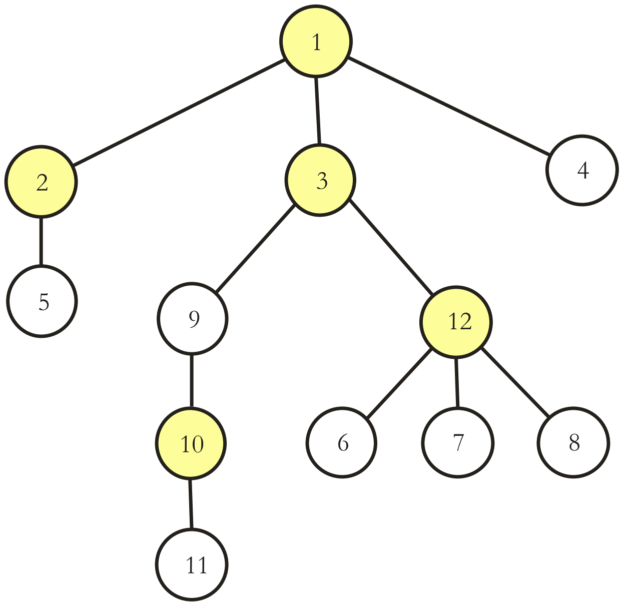
Készíts programot, amely kiszámítja, hogy legkevesebb hány dolgozónak kell részt vennie képzésen, és meg is adja, hogy kiknek!

A bemenet első sorában a dolgozók száma van (`1≤N≤100 000`). Az igazgató azonosítója 1. A második sor pontosan N egész számot tartalmaz, az i-edik szám az i azonosítójú dolgozó közvetlen főnökének azonosítója. Mivel az igazgatónak nincs főnöke, ezért az első szám 0.

A kimenet első sorába azt az M számot kell írni, ahány dolgozónak rész kell vennie kiképzésen! A második sorba M számot kell írni, azon dolgozók sorszámait, akik részt vesznek kiképzésen! A számok sorrendje közömbös. Több megoldás esetén bármelyik megadható.

Példa:

Bemenet:	Kimenet:
12	5
0 1 1 1 2 12 12 12 3 9 10 3	1 2 3 10 12



5. feladat: Csoport beosztás (15 pont)

Egy kiránduláson részvevő tanulókból két csapatot kell képezni! A két csapatot úgy kell képezni, hogy ha X és Y barátok, akkor azonos csapatba kerüljenek, de ha nem kedvelik egymást, akkor nem kerülhetnek egy csapatba!

Készíts programot, amely kiszámít egy, a feltételeknek megfelelő csapatbeosztást!

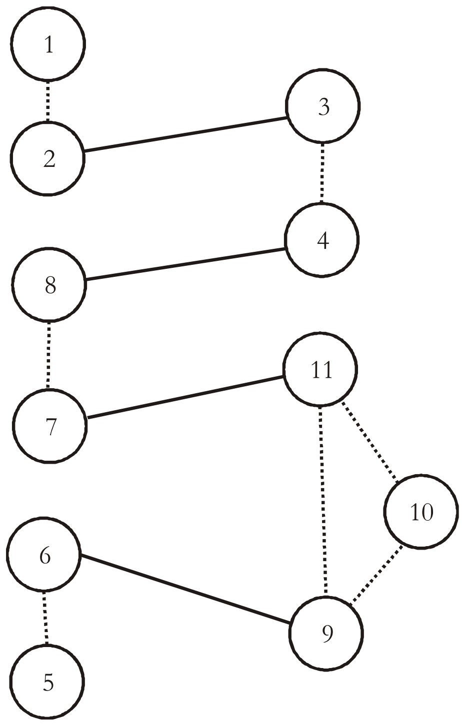
A bemenet első sorában a tanulók száma (`1≤N≤500`) és a baráti párok száma (`1≤M≤20000`) és azon párok K száma van, akik nem kedvelik egymást. A következő M sor a baráti párokat, az azt követő K sor pedig azon párokat tartalmazza, akik nem kedvelik egymást.

A kimenet első sorába az egyik, a második sorába a másik csapatba sorolt tanulók azonosítóit kell írni, tetszőleges sorrendben! Több megoldás esetén bármelyik megadható. Ha nincs megoldás, akkor a kimenet első és egyetlen sorába a -1 számot kell írni!

Példa:

```
Bemenet:
11 7 4
1 2
3 4
5 6
9 10
10 11
11 9
7 8
2 3
6 9
8 4
7 11
```

```
Kimenet:
 1 2 5 6 7 8
3 4 9 10 11
```



2011. A verseny végeredménye

I. korcsoport

1	Alexy Marcell	Juhász Gyula Általános Iskola, Vác
	Weisz Ambrus	Fazekas Mihály Gimnázium, Budapest
3	Leitereg Miklós	Veres Péter Gimnázium, Budapest
4	Almási Péter Béla	Fazekas Mihály Gimnázium, Debrecen
5	Horváth István	Fazekas Mihály Gimnázium, Debrecen
	Erdős Márton	Batthyány Lajos Gimnázium és EüSzKI, Nagykanizsa
	Székely Gábor	Fazekas Mihály Gimnázium, Debrecen
8	Hoffmann Áron	Vörösmarty Mihály Gimnázium, Érd
9	Volford András	Radnóti Miklós Gimnázium, Szeged
10	Jákli Aida	Zrínyi Miklós Gimnázium, Zalaegerszeg

II. korcsoport

1	Szabó Attila	Leőwey Klára Gimnázium, Pécs
2	Weisz Gellért	Fazekas Mihály Gimnázium, Budapest
3	Nagy Róbert	Fazekas Mihály Gimnázium, Budapest
	Barta János	Madách Imre Gimnázium, Salgótarján
5	Virág Fausztin Asztrik	Földes Ferenc Gimnázium, Miskolc
	Székely Szilveszter	Neumann János Középiskola és Kollégium, Eger
7	Nagy Vendel	Fazekas Mihály Gimnázium, Debrecen
8	Borsos Tamás	Bolyai Farkas Elméleti Líceum, Marosvásárhely
	Nemkin Viktória	Fazekas Mihály Gimnázium, Debrecen
10	Palkó András	Vörösmarty Mihály Gimnázium, Szentgotthárd

III. korcsoport

1	Szenczi Zoltán	Fazekas Mihály Gimnázium, Budapest
2	Palincza Richárd Péter	Berzsenyi Dániel Gimnázium, Budapest
3	Marussy Kristóf	Szent István Gimnázium, Budapest
4	Dankovics Attila János	Veres Péter Gimnázium, Budapest
5	Sulyok András Attila	Református Gimnázium, Szentendre
6	Kovács Gábor Ferenc	Árpád Gimnázium, Tatabánya
7	Radnai Balázs	Fazekas Mihály Gimnázium, Budapest
8	Dénes György	Radnóti Miklós Gimnázium, Dunakeszi
9	Erdős Gergely	Batthyány Lajos Gimnázium és Szakközépiskola, Nagykanizsa
10	Berghammer Tamás	Illyés Gyula Gimnázium és Szakközépiskola, Budaörs
11	Danyluk Tamás	Földes Ferenc Gimnázium, Miskolc
12	Kerekes Dávid	Premontrei Szent Norbert Gimnázium, Gödöllő

2012. Első forduló

Ötödik-nyolcadik osztályosok

Számítógép nélküli feladatok

1. feladat: Hibakeresés (24 pont)

Az alábbi eljárás az N elemű T tömbben kapott keresztnevek közül meghatározná, hogy hány különböző van köztük (Db) és melyek ezek (K). Az algoritmusba több hiba került, ami miatt hibásan működik. Melyek ezek?

```
Keresztnevek(N,T,Db,K):
  K(Db):=T(1); Db:=1
  Ciklus i=2-től N-ig
    j:=i
    Ciklus amíg j<Db és T(j)=K(i)
      j:=j+1
    Ciklus vége
    Ha j≤Db akkor Db:=Db+1; K(Db):=i
  Ciklus vége
Eljárás vége.
```

2. feladat: Mit csinál (26 pont)

Az alábbi algoritmus az N elemű T vektor alapján számolja ki K, V és A értékét (N>1).

```
Valami(N,T,K,V,A):
  A:=0; B:=0; j:=1; K:=0; V:=0
  Ciklus i=1-től N-ig
    Ha T(i)<0 akkor Ha B>A akkor K:=j; V:=i-1; A:=B
                    j:=i+1; B:=0
    különben B:=B+T(i)
  Ciklus vége
  Ha B>A akkor K:=j; V:=N; A:=B           {*}
Eljárás vége.
```

A. Mi lesz K, V és A értéke, ha N=3, T =(-5,3,2)?

B. Mi lesz K, V és A értéke, ha N=6, T =(-5,3,-1,2,6,0)?

C. Fogalmazd meg általánosan, hogy mi lesz az eljárás végén K, V és A!

D. Milyen T vektor esetén marad A értéke 0?

E. Milyen T vektor esetén hajtódik végre a *-gal jelölt elágazás akkor-ága?

F. Milyen T vektor esetén marad j értéke 1?

3. feladat: Kitaláló (20 pont)

Az alábbi algoritmus 4 paramétert kap (A,B,C,D), amelyekből egy értéket számol ki (E), mindegyik paraméter szöveg típusú.

Az A és B értéke ”a”, ”b” és ”0” lehet, C és D értéke pedig ”+ ” vagy ”-”.

```
Valami(A,B,C,D,E):
  Ha A=”a” vagy B=’a’ akkor p:=igaz különben p:=hamis
  Ha A=”b” vagy B=’b’ akkor q:=igaz különben q:=hamis
  Ha C=”+” vagy D=’+’ akkor r:=igaz különben r:=hamis
  Ha p akkor E:=”A”
            Ha q akkor E:=E+”B”
  különben Ha q akkor E:=”B”
            különben E:=”0’
  Ha r akkor E:=E+”+” különben E:=E+”-”
Eljárás vége.
```

Készíts táblázatot, amely az összes bemenetre megadja az E eredmény értékét!

Számítógépes feladat – VÁLASZTHATÓ

4. feladat: Időjárás (30 pont)

Ismerjük a következő N napra az időjárás előrejelzést: minden napra a várható legkisebb és legnagyobb hőmérsékletet, mindkettő egész szám.

Készíts programot, amely

A. megadja azon napok számát, ahol a várható maximum legalább 10 fokkal több, mint a várható minimum;

B. megad egy napot, amikor a várható minimum nagyobb, mint a másnapra várt maximum; ha nincs ilyen, akkor 0 az eredmény

C. megadja azt a napot, ahol a várható maximum és minimum eltérése a lehető legkisebb!

Számítógép nélküli feladat – VÁLASZTHATÓ

4. feladat: Időjárás (30 pont)

Ismerjük a következő N napra az időjárás előrejelzést: minden napra a várható legkisebb és legnagyobb hőmérsékletet, az N elemű Max, illetve Min vektorokban. Vegyük minden napra napi átlagnak az előrejelzésben szereplő aznapi legkisebb és legnagyobb érték átlagát!

Meg kellene adni:

A. az N napra várt legmagasabb hőmérsékletet;

B. egy olyan napot, amikor fagyni is fog, de olvadás is lesz;

C. azt a napot, amelyre a legalacsonyabb átlag várható;

D. olyan napi átlagok átlagát, amelyek magasabbak az előző nap maximumánál és alacsonyabbak a következő nap minimumánál!

Papíron az alábbi algoritmust kaptuk, egyes helyeken azonban nem látszik az írás. Pótold a hiányos helyeken az algoritmust, hogy az a feladat helyes megoldása legyen!

```
Mérések(N,X):
  a:=Max(1); b:=0; c:=1; volt:=hamis; dbd:=0; sd:=0
  Ciklus i=2-től N-ig
    Ha □□□□□□□□□□□□□□□□ akkor a:=Max(i)
    Ha □□□□□□□□□□□□□□□□ akkor c:=i
  Ciklus vége
  Ciklus i=1-től N-ig
    Ha □□□□□□□□□□□□□□□□ akkor volt:=igaz; b:=i
  Ciklus vége
  Ciklus i=2-től N-1-ig
    Ha □□□□□□□□□□□□□□□□ akkor dbd:=dbd+1; sd:=sd+□□□□□□
  Ciklus vége
  Ha □□□□□□□ akkor d:=sd/dbd
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Időjárás (30 pont)

Az ország N településére kaptuk meg M napra az időjárás-előrejelzést, a H(N,M) mátrixban, ahol H(i,j) az i-edik településen a j-edik napra várható maximális hőmérséklet. Az alábbi algoritmus megadná azokat a településeket, ahol folyamatosan legalább K egymást követő napon lesz F fok feletti a hőmérséklet, ha jó lenne.

A. Jelöld be, mik a hibák benne!

B. Mi az eredmény és melyik változókban van?

C. Mi a szerepe az A változónak?

```
Forró(N,M,H,K,F):
  D:=0
  Ciklus i=1-től M-ig
    A:=0; j:=1
    Ciklus amíg j≤M vagy A≤K
      Ha H(i,j)>F akkor A:=A+1 különben D:=0
      j:=i+1
    Ciklus vége
    Ha A<K akkor D:=D+1; Y(D):=j
  Ciklus vége
Eljárás vége.
```

2. feladat: Gyorsabbra (20 pont)

Az alábbi algoritmus megadja a leghosszabb folytonos szakaszt az N elemű T tömbben, ahol csak prímszám van. A T tömb 1 és M közötti egész számokat tartalmaz. Feltehető, hogy N sokkal nagyobb, mint M. A H változó a szakasz hossza lesz, a W pedig igaz lesz, ha van olyan szakasz, ahol legalább 1 prímszám van.

Írd át hatékonyabbra (gyorsabbra) és magyarázd is a megoldásod!

```
Szakasz(N,T,H,W):
  H:=0; i:=1; W:=hamis
  Ha prím(T(1)) akkor Db:=1
  Ciklus i=2-től N-ig
    Ha prím(T(i)) akkor Ha prím(T(i-1)) akkor Db:=Db+1
                        különben Db:=1
    különben Ha Db>H akkor H:=Db; W:=igaz
  Ciklus vége
  Ha Db>H akkor H:=Db; W:=igaz
Eljárás vége.
```

```
prím(x):
  j:=2
  Ciklus amíg j<x és j nem osztója x
    j:=j+1
  Ciklus vége
  prím:=j=x
Függvény vége.
```

3. feladat: Munka (25 pont)

Egy vállalkozó megrendeléseket fogad. Minden megrendelt munkát pontosan egy nap alatt tud elvégezni. Minden megrendelés tartalmazza, hogy az adott munkát mikorra kell elvégezni, és mekkora hasznot eredményez, ha a munkát határidőig elvégzi a vállalkozó. A vállalkozó a megrendelések közül ki akarja választani azokat, amelyeket el tud végezni határidőre és a lehető legtöbb hasznot adják.

Add meg, hogy az alábbi megrendelések esetén mennyi lehet a haszon és ehhez mely munkákat kell elvégezni!

A. N=8, munkák=(2,1),(2,3),(2,8),(3,7),(3,10),(4,5),(4,11),(5,4)
magyarázat: határidő a 2. nap, összeg 1 forint, ..., határidő az 5. nap, összeg 4 forint.

B. N=7, munkák=(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7)

C. N=7, munkák=(2,1),(3,2),(3,3),(4,4),(4,5),(5,6),(5,7)

D. N=7, munkák=(2,9),(3,2),(3,4),(4,3),(4,6),(5,5),(5,7)

E. N=9, munkák=(2,9),(3,9),(3,9),(4,3),(4,6),(4,9),(5,5),(5,7),(5,9)

4. feladat: Kitaláló (25 pont)

Egy N elemű T tömbben egész számok vannak. Kezdetben a tömb minden elemére igaz, hogy T(i)≤T(2*i) és T(i)≤T(2*i+1), feltéve hogy 2*i≤N, illetve 2*i+1≤N).

Két eljárást írtunk:

```
Egyik(x):
  N:=N+1; T(N):=X; i:=N
  Ciklus amíg i>1 és T(i)<T(i div 2)
    Csere(T(i div 2),T(i)); i:=i div 2
  Ciklus vége
Eljárás vége.
```

```
Másik(x):
  x:=T(1); T(1):=T(N); N:=N-1; i:=1
  Ciklus amíg 2*i≤N
    j:=2*i
    Ha j<N és T(j+1)<T(j) akkor j:=j+1
    Ha T(i)>T(j) akkor Csere(T(i),T(j)); i:=j különben i:=N+1
  Ciklus vége
Eljárás vége.
```

A. Add meg az Egyik eljárás minden cikluslépésében, hogy a ciklus feltételének vizsgálatakor mi a T tömb elemeinek értéke, ha kezdetben N=7, T=(1,3,6,5,4,7,8) és x=2!

B. Add meg a Másik eljárás minden cikluslépésében, hogy a ciklus feltételének vizsgálatakor mi a T tömb elemeinek értéke, ha kezdetben N=8, T=(1,2,6,3,4,7,8,5) és mennyi lesz x értéke!

C. Milyen feltétel teljesül a Csere művelet után az Egyik eljárásban T(i),T(2*i) és T(2*i+1) értékére?

D. Milyen feltétel teljesül a Csere művelet után a Másik eljárásban T(i),T(2*i) és T(2*i+1) értékére?

E. A Másik eljárásban a j index értéke a Csere művelet kezdetekor hogyan függ a T tömb elemeitől?

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Időjárás (20 pont)

Az ország N településére kaptuk meg az M napos időjárás-előrejelzést, a H(N,M) mátrixban, ahol H(i,j) az i-edik településen a j-edik napra várható maximális hőmérséklet. Az alábbi algoritmus megadná a legszélsőségesebb településeket, azaz azokat, ahol a legkisebb és a legnagyobb várt hőmérséklet eltérése maximális.

A. Jelöld be, mik a hibák benne!

B. Mi az eredmény és melyik változókban van?

C. Mi a szerepe az A és a B változónak?

```
Szélsőséges(N,M,H):
  C:=0
  Ciklus i=1-től N-ig
    A:=H(i,1); B:=H(i,M)
    Ciklus j=2-től M-ig
      Ha H(i,j)>A akkor A:=H(i,j)
      Ha H(i,j)<B akkor A:=H(j,i)
    Ciklus vége
    Ha A-B>C akkor D:=D+1; Y(D):=i; C:=A-B
    különben ha A-B=D akkor D:=D+1; Y(B):=i
  Ciklus vége
Eljárás vége.
```

2. feladat: Gyorsabbra (20 pont)

Az alábbi algoritmus megadja azt a H hosszú szakaszt az N elemű T tömbben, ahol a legtöbb prímszám van. A T tömb 1 és M közötti egész számokat tartalmaz. Feltehető, hogy N sokkal nagyobb, mint M. A K változó a szakasz kezdete lesz, a Van pedig igaz lesz, ha van olyam szakasz, ahol legalább 1 prímszám van.

Írd át hatékonyabbra (gyorsabbra) és magyarázd is a megoldásod!

```
Szakasz(N,T,H,K,Van):
  D:=0; Van:=hamis
  Ciklus i=1-től N-H+1-ig
    Db:=0
    Ciklus p=i-től i+H-1-ig
      Ha prím(T(p)) akkor Db:=Db+1
    Ciklus vége
    Ha Db>D akkor K:=i; D:=Db; Van:=igaz
  Ciklus vége
Eljárás vége.
```

```
Prím(x):
  j:=2
  Ciklus amíg j<x és j nem osztója x
    j:=j+1
  Ciklus vége
  prím:=j=x
Függvény vége.
```

3. feladat: Taxi (20 pont)

Egy taxis vállalkozó N megálló között szállít utasokat minibusszal. Egy menetben mindig az 1. megállótól indul és az i-edik megállótól (i<N) az i+1-edik megállóba kell mennie. Ismeri az utasok

igényeit, tehát minden utasról tudja, hogy melyik megállótól melyik megállóig akar utazni. A taxin egyszerre legfeljebb K utas utazhat.

Számold ki az alábbi esetekre, hogy legjobb esetben összesen hány utast tud egy menetben az utas igényének megfelelő helyre elszállítani és add meg, hogy melyikeket!

A. K=3, az igények: (1,7), (2,3), (2,3), (2,3), (3,5), (3,5), (4,7), (6,8).

B. K=2, az igények: (1,6), (2,4), (3,6), (4,5), (5,8), (6,7)

C. K=3, az igények: (2,3),(2,5),(2,5),(3,5),(3,6),(3,9),(4,6),(5,10),(6,10),(7,9),(8,9),(9,10)

D. K=4, az igények: (2,3),(2,5),(2,5),(3,5),(3,6),(3,9),(4,6),(5,10),(6,10),(7,9),(8,9),(9,10)

<u>4. feladat</u>: Kitaláló (20 pont)

Egy N elemű T tömbben egész számok vannak. Kezdetben a tömb minden elemére igaz, hogy T(i)≤T(2*i) és T(i)≤T(2*i+1), feltéve hogy 2*i≤N, illetve 2*i+1≤N).

Két eljárás-párt írtunk: Egyik(x), A(i) és Másik(x), B(i)

```
Egyik(x):
  N:=N+1; T(N):=X; A(N)
Eljárás vége.
A(i):
  Ha i>1 és T(i)<T(i div 2)
    akkor Csere(T(i div 2),T(i)); A(i div 2)
Eljárás vége.
Másik(x):
  x:=T(1); T(1):=T(N); N:=N-1; B(1)
Eljárás vége.
B(i)
  Ha 2*i≤N akkor j:=2*i
               Ha j<N és T(j+1)<T(j) akkor j:=j+1
               Ha T(i)>T(j) akkor Csere(T(i),T(j)); B(j)
  Elágazás vége
Eljárás vége.
```

A. Add meg az A eljárás minden hívásában, hogy az elágazás feltételének vizsgálatakor mi a globális T tömb elemeinek értéke, ha kezdetben T=(1,3,6,5,4,7,8) és x=2!

B. Add meg a B eljárás minden hívásában, hogy a külső elágazás feltételének vizsgálatakor mi a globális T tömb elemeinek értéke, ha kezdetben T=(1,2,6,3,4,7,8,5) és mennyi lesz x értéke!

C. Milyen feltétel teljesül a Csere művelet után az A eljárásban T(i), T(2*i) és T(2*i+1) értékére?

D. Milyen feltétel teljesül a Csere művelet után a B eljárásban T(i), T(2*i) és T(2*i+1) értékére?

E. A B eljárásban a j index értéke a Csere művelet kezdetekor hogyan függ a globális T tömb elemeitől?

<u>5. feladat</u>: Képátló (20 pont)

Adott egy N×N pixelből álló fekete-fehér kép. Szeretnénk a képen a bal felső saroktól a jobb alsó sarokig egy jobbra-lefele haladó határvonalat húzni úgy, hogy a vonaltól jobbra-felfele eső fekete (0 értékű), valamint a vonaltól balra-lefele eső fehér (1 értékű) pixelek számának K összege a lehető legkevesebb legyen. A határvonalra eső pixelek nem számítanak bele.

Add meg, a megoldást az alábbi bemenetekre!

A.
```
1 1 1 1 1
0 1 1 1 0
1 0 0 1 1
0 1 1 1 0
0 1 1 0 0
```

B.
```
1 1 1 1 1
1 1 1 1 1
0 0 1 1 1
0 0 1 1 0
0 0 1 1 0
```

C.
```
1 1 1 1 1
0 1 1 1 0
0 0 0 1 1
0 0 1 0 0
0 1 1 0 1
```

D. Add meg azt a T[i,j] függvényt, ami az (i,j) ponttól jobbra lefelé adja a megoldást! Segédfüggvényeket használhatsz hozzá.

2012. Második forduló

Ötödik-nyolcadik osztályosok

1. feladat: Időjárás (30 pont)

Megkaptuk egy település elmúlt N napon délben mért hőmérsékletét.

Készíts programot, amely beolvassa a napok számát (`1≤N≤100`), egy hőmérséklet értéket (`-20≤K ≤50`), valamint az egyes napokon mért hőmérsékletet (`-20≤H(i)≤50`), majd megadja

A. a legmelegebb napot;

B. a K foknál melegebb napok számát;

C. a leghosszabb időszakot, amikor folyamatosan K fok felett volt a hőmérséklet!

Példa:

```
Bemenet:
Napok száma: 7
Korlát: 30
1. nap: 25
2. nap: 36
3. nap: 29
4. nap: 33
5. nap: 34
6. nap: 36
7. nap: 30
```

```
Kimenet:
Legmelegebb: 2. nap
30 foknál több: 4 nap
Leghosszabb meleg időszak: 4-6. nap
```

2. feladat: Tükörkép számok (16 pont)

Írj programot, amely megadja azokat a 2 számjegyű számokat, amelyeket tükörképükkel megszorozva olyan számot kapunk, amely saját magának tükörképe!

Példa:

```
Kimenet:
…
12: 12*21=252
…
```

3. feladat: Életjáték (29 pont)

A N×N-es négyzetrács mezőit celláknak, a korongokat sejteknek nevezzük. Egy cella környezete a hozzá legközelebb eső 8 mező (tehát a cellához képest „átlósan” elhelyezkedő cellákat is figyelembe vesszük). Egy sejt/cella szomszédjai a környezetében lévő sejtek. A játék körökre osztott, a kezdő állapotban tetszőleges számú (egy vagy több) cellába sejteket helyezünk. Ezt követően a játékosnak nincs beleszólása a játékmenetbe. Egy sejttel (cellával) egy körben a következő három dolog történhet:

A sejt túléli a kört, ha két vagy három szomszédja van.

A sejt elpusztul, ha kettőnél kevesebb (elszigetelődés), vagy háromnál több (túlnépesedés) szomszédja van.

Új sejt születik minden olyan cellában, melynek környezetében pontosan három sejt található.

Fontos, hogy a változások csak a kör végén következnek be, tehát az „elhalálozók” nem akadályozzák a születést és a túlélést (legalábbis az adott körben), és a születések nem mentik meg az „elhalálozókat”.

Készíts programot, amely beolvassa a négyzetrács méretét (`2≤N≤10`), a kezdetben felteendő sejtek számát (`2≤M≤10`), a lépések számát (`1≤L≤100`), valamint a sejtek sor- és oszlopindexét (`1≤S(i),O(i)≤N`), majd lejátssza a játékot L lépésen át!

```
Bemenet:

Méret: 6
Sejtek száma: 5
1. sejt helye: 4 4
2. sejt helye: 5 4
3. sejt helye: 6 4
4. sejt helye: 4 5
5. sejt helye: 5 6
Lépések száma: 4
```

`Kimenet:`

`Kiindulás:`

			X	X	
			X		X
			X		

`1. lépés:`

			X	X	
		X	X		
				X	

`2. lépés:`

		X	X	X	
		X			
			X		

`3. lépés`

			X		
		X	X		
		X		X	

`4. lépés:`

		X	X		
		X		X	
		X			

Kilencedik-tizedik osztályosok

<u>1. feladat</u>: Előrejelzés (16 pont)

Ismerjük N településre az M napos időjárás-előrejelzést, ezek alapján keressük a legmelegebb települést. Készíts programot, amely megadja négyféle értelmezés szerint a legmelegebb települést:

A. a legmelegebb település az, amelyre az előrejelzések maximuma a legnagyobb;

B. a legmelegebb település az, amelyre az előrejelzések átlaga a legnagyobb;

C. a legmelegebb település az, amelyben a leghosszabb időszakon belül várható folyamatosan K fok feletti hőmérséklet;

D. a legmelegebb település az, amelyre a legtöbb napon fordul elő, hogy a várt hőmérséklet nagyobb minden más arra a napra előrejelzett hőmérsékletnél!

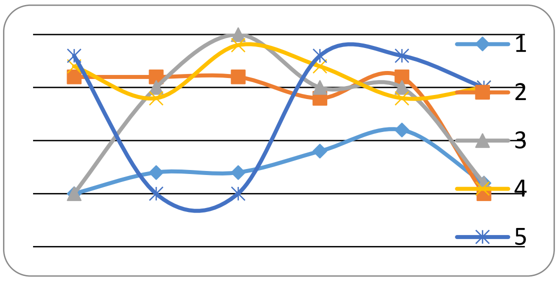
Az bemenet első sorában a települések száma (`1≤N≤1000`), a napok száma (`1≤M≤1000`) és a hőmérséklet korlát van (`20≤K≤50`). A következő N sor mindegyikében M egész szám van, az i-edik település j-edik napra várt hőmérséklete.

A kimenet négy sorába egy-egy település sorszámát kell írni! Az első sorba az A, a másodikba a B, a harmadikba a C, a negyedikbe pedig a D szempont szerinti legmelegebb települést! Ha több megoldás van, bármelyik megadható. Ha nincs megoldás (C és D részfeladatban), akkor -1-et kell kiírni!

<u>Példa</u>:

```
Bemenet:
5 6 30
20 22 22 24 26 21
31 31 31 29 31 20
20 30 35 30 30 21
32 29 34 32 29 30
33 20 20 33 33 30
```

```
Kimenet:
3
4
2
5
```



<u>2. feladat</u>: Verseny (20 pont)

Egy kieséses versenyben ismerjük a csapatok mérkőzéseit: ki kit győzött le. Írj programot, amely megadja:

A. a még versenyben levőket;

B. azokat a csapatokat, amelyek legalább egyszer győztek, de már kiestek;

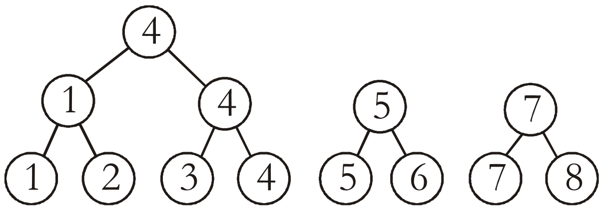
C. a legtöbb csapatot közvetlenül vagy közvetve legyőző csapatot!

A bemenet első sorában a csapatok száma (`2≤N≤1000`) és a mérkőzések száma van (`1≤M<N`). A következő M sor mindegyikében két csapat sorszáma van (`1≤I≠J≤N`), ami azt jelenti, hogy az I-edik csapat legyőzte a J-edik csapatot.

A kimenet első sorába a még versenyben levő csapatok darabszámát, majd a sorszámát kell írni (növekvő sorrendben), a második sorba azok darabszámát, majd a sorszámát, amelyek úgy estek ki, hogy legalább egyszer győztek (növekvő sorrendben), a harmadik sorba pedig azt a csapatot, amely a legtöbb más csapatot győzte le közvetve vagy közvetlenül! Ha több megoldás van, bármelyik kiírható.

Példa:

```
Bemenet:          Kimenet:
8 5               3 4 5 7
1 2               1 1
4 3               4
4 1
7 8
5 6
```



3. feladat: Könyvtári pakolás (18 pont)

Egy könyvtár polcán egy sorban N darab könyv van, de nem a kívánt sorrendben. A könyvtáros minden könyvre ráragasztott egy cetlit, amire ráírta, hogy a helyes sorrendben hányadik helyen kell majd lennie. A kívánt sorrend kialakítását párok cseréjével akarja megvalósítani.

Készíts programot, amely kiszámítja, hogy legkevesebb hány cserével lehet kialakítani a kívánt sorrendet!

A bemenet első sorában a könyvek száma van (2≤N≤30 000). A második sor pontosan N különböző pozitív egész számot tartalmaz, az i-edik szám értéke az i-edik könyv helyes sorrendbeli sorszáma.

A kimenet első és egyetlen sora azt a legkisebb M számot tartalmazza, amire a bemenetben megadott könyvsorozat M számú cserével a kívánt sorrendbe rakható!

Példa:

```
Bemenet:                 Kimenet:
10                       7
7 10 1 3 2 8 4 9 6 5
```

4. feladat: Számjáték (21 pont)

Tekintsük a következő egyszemélyes játékot: A játék kezdetén egy sorban leraknak N darab pozitív egész számot. A játékos legfeljebb K lépést tehet. Egy lépésben a még a táblán lévő számsorból két egymás melletti számot levehet, a levett számok a pontszámához adódnak. A levett számok helye üresen marad, és a lépés során a szomszédos számok között nem lehet üres hely. A játékosnak az a célja, hogy a lehető legtöbb pontot szerezze.

Készíts programot, amely kiszámítja, hogy legjobb esetben hány pontot szerezhet a játékos!

A bemenet első sorában a kezdeti számsorozat számainak száma (1≤N≤10 000), és a lépések maximális száma (1≤K≤1000) van. A második sor tartalmazza a kezdeti játékállást (minden szám értéke legfeljebb 5000).

A kimenet egyetlen sora a játékban elérhető lehető legtöbb pont értékét tartalmazza!

Példa:

```
Bemenet:                 Kimenet:
6 2                      27
1 [6 8] [7 6] 2
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Előrejelzés (15 pont)

Ismerjük N településre az M napos időjárás-előrejelzést, ezek alapján keressük településeket.

Készíts programot, amely megad négyféle értelmezés szerint egy-egy települést:

A. azt a települést, amelyre a legnagyobb és a legkisebb előrejelzés eltérése a legnagyobb;

B. azt a települést, amelyre van olyan település, ahol minden nap nála hidegebb várható;

C. egy olyan települést, amelyben a leghosszabb időszakon belül várható folyamatosan K fok feletti hőmérséklet;

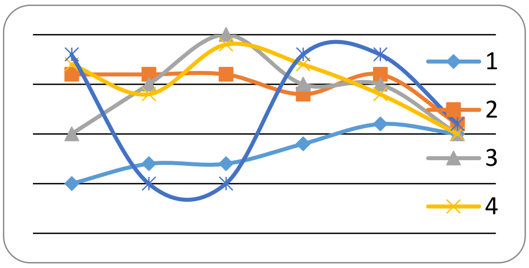
D. azt a települést, amelyre a legtöbb napon fordul elő, hogy a várt hőmérséklet nagyobb minden más, arra a napra előrejelzett hőmérsékletnél.

A bemenet első sorában a települések száma (1≤N≤1000), a napok száma (1≤M≤1000) és a hőmérséklet korlát van (20≤K≤50). A következő N sor mindegyikében M egész szám van, az i-edik település j-edik napra várt hőmérséklete.

A kimenet négy sorába egy-egy település sorszámát kell írni! Az első sorba az A, a másodikba a B, a harmadikba a C, a negyedikbe pedig a D értelmezés szerinti települést! Ha több megoldás van, bármelyik megadható. Ha nincs megoldás (B,C,D részfeladatban), abba a sorba -1-et kell írni!

Példa:

Bemenet:	Kimenet:
5 6 30	5
20 22 22 24 26 25	2
31 31 31 29 31 26	2
25 30 35 30 30 25	5
32 29 34 32 29 25	
33 20 20 33 33 26	



2. feladat: Verseny (15 pont)

Egy kieséses versenyben ismerjük a csapatok mérkőzéseit: ki kit győzött le.

Írj programot, amely megadja:

A. azt a csapatot, amely a kiesettek közül a legtöbbször győzött;

B. a legtöbb csapatot közvetlenül vagy közvetve legyőző csapatot;

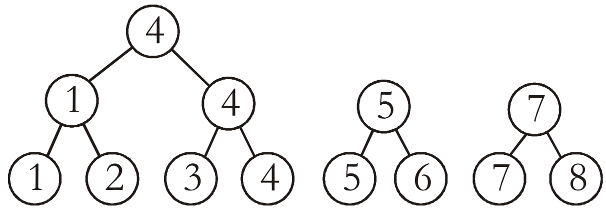
C. a következő mérkőzést játszó két csapatot, amely két olyan versenyben levő csapat legyen, amely eddig közvetve vagy közvetlenül a lehető legkevesebb csapatot győzte le!

A bemenet első sorában a csapatok száma (2≤N≤1000) és a mérkőzések száma van (1≤M<N). A következő M sor mindegyikében két csapat sorszáma van (1≤I≠J≤N), ami azt jelenti, hogy az I-edik csapat legyőzte a J-edik csapatot.

A kimenetelső sorába a kiesettek közül legtöbb győzelmet szerző csapat sorszámát kell írni (-1-et, ha nincs ilyen csapat)! A második sorba azt a csapatot, amely a legtöbb más csapatot győzte le közvetve vagy közvetlenül, a harmadik sorba a szabály szerint a következő mérkőzést játszó két csapat sorszámát! Ha több megoldás van, bármelyik kiírható, a harmadik sorban az egyetlen -1 szám álljon, ha nincs megoldás!

Példa:

Bemenet:	Kimenet:
8 5	1
1 2	4
4 3	5 7
4 1	
7 8	
5 6	



3. feladat: Üvegválogatás (15 pont)

Egy üzletben válogatás nélkül gyűjtötték össze az üres üvegeket N ládában. Mivel az üvegfajták száma is N, ezért az azonos fajtájú üvegeket egy-egy ládában el lehet tárolni. Ezért szét akarják válogatni az üvegeket, hogy minden ládában csak azonos fajtájú legyen, de úgy, hogy a lehető legkevesebb üveget kelljen átrakni más ládába.

Készíts programot, amely kiszámítja, hogy legkevesebb hány darab üveget kell másik ládába átrakni, és megadja, hogy ehhez az egyes fajtákat melyik ládába kell gyűjteni!

A bemenet első sorában egy egész szám, a ládák száma van (1≤N≤8). A következő N sor mindegyike pontosan N egész számot tartalmaz, közülük az I-edik sorban a J-edik szám az I-edik ládában lévő J-fajta üvegek száma. Minden szám értéke legfeljebb 5000.

A kimenet első sora a kívánt szétválogatáshoz szükséges átrakások minimális számát tartalmazza! A második sor pontosan N egész számot tartalmazzon: az I-edik szám annak az üvegfajtának a sorszáma legyen, amelyet az I-edik ládába rakunk!

Példa:

Bemenet:	Kimenet:
3	182
15 30 8	3 2 1
55 80 10	
50 60 12	

4. feladat: Párosítás (15 pont)

Egy raktárból N boltba kell kiszállítani ládákba csomagolt árut. Minden boltba pontosan két ládát kell vinni. A ládák az előkészítés időrendi sorrendjében egymás mellett egy sorban vannak, mindegyikre ráragasztva annak a boltnak a sorszáma, ahova szállítani kell. A raktárosnak át kell rendezni a ládák sorrendjét, hogy az egy boltba kerülő ládák egymás mellett legyenek. Az átrendezés során egy lépésben két láda helyét cserélheti meg.

Készíts programot, amely kiszámítja, hogy legkevesebb hány cserével lehet kialakítani a kívánt sorrendet!

A bemenet első sorában a boltok száma van (2≤N≤20 000). A második sor pontosan 2*N bolt sorszámot tartalmaz, az 1,…,N számok mindegyike pontosan kétszer fordul elő a sorban.

A kimenet első és egyetlen sora azt a legkisebb M számot tartalmazza, amire a bemenetben megadott ládasor M számú cserével a kívánt sorrendbe rakható!

Példák:

Bemenet:	Kimenet:	Magyarázat:
4 1 3 2 1 3 4 4 2	3	1 3 2 1 3 4 4 2 1 1 2 3 3 4 4 2 1 1 2 3 3 2 4 4 1 1 2 2 3 3 4 4

Bemenet:	Kimenet:	Magyarázat:
4 3 2 1 4 2 3 1 4	2	3 2 1 4 2 3 1 4 3 2 1 1 2 3 4 4 2 2 1 1 3 3 4 4

5. feladat: Számjáték (15 pont)

Tekintsük a következő egyszemélyes játékot: A játék kezdetén egy sorban leraknak N darab pozitív egész számot. A játékos legfeljebb L lépést tehet. Egy lépésben a még a táblán lévő számsorból H darab egymás melletti számot levehet, a levett számok a pontszámához adódnak. A levett számok helye üresen marad, és lépés során a szomszédos számok között nem lehet üres hely. A játékosnak az a célja, hogy a lehető legtöbb pontot szerezze.

Készíts programot, amely kiszámítja, hogy legjobb esetben hány pontot szerezhet a játékos!

A bemenet első sorában a kezdeti számsorozat számainak száma ($1 \le N \le 3000$), a lépések maximális száma ($1 \le L \le 1000$) és az egyszerre levehető számok darabszáma ($2 \le H \le N$) van. A második sor tartalmazza a kezdeti játékállást (minden szám értéke legfeljebb 5000).

A kimenet első sora a játékban elérhető lehető legtöbb pont értékét tartalmazza! A második sor egy olyan lépéssort tartalmazzon, amellyel a maximális pontszám elérhető! Egy lépés a lépésben levett számsor első elemének sorszáma legyen!

Példa:

Bemenet:	Kimenet:
8 2 3 1 $\boxed{6\ 8\ 7}$ 6 $\boxed{2\ 1\ 8}$	32 2 6

2012. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Állat (25 pont)

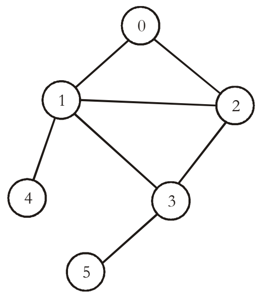
Egy állatkertben ismerjük a bejárható útvonalakat. A bejárat a 0 sorszámú pont. Az egyes állatokat az 1 és N közötti sorszámukkal azonosítjuk ($1 \le N \le 100$), az utakat pedig két olyan állat sorszámával, amelyek ketrece között vezetnek.

Készíts programot, amely az állatkerti utak ismeretében megadja, hogy hány olyan állat van, amelyik zsákutca végén található, valamint azt, hogy melyik állathoz vezet a legtöbb út (ha több is van, bármelyik megadható)!

Példa:

```
Bemenet:
Állatok száma: 5
Utak száma: 7
1. út: 0 1
2. út: 1 4
3. út: 3 1
4. út: 3 5
5. út: 2 0
6. út: 2 3
7. út: 1 2
```

```
Kimenet:
Állatok száma zsákutca végén: 2
Legtöbb út: 1
```



2. feladat: Titkosírás (25 pont)

Egy titkosírás elkészítéséhez a következő táblázatot használjuk:

	a	b	c	d	e	f	g	h	i
a	i	o	q	h	b	f	y	l	w
b	n	r	a	g	s	k	t	e	z
c	d	u	p	x	c	j	v	m	

Egy szót ezzel a táblázattal úgy titkosítunk, hogy a betűit egyesével megkeressük a táblázat belsejében és a titkosított szövegbe a helyére a betű sorában, illetve oszlopában levő betűpárt tesszük. Feltehetjük, hogy csak az angol ábécé kisbetűit használjuk.

Példa:

```
A titkosítandó szó: balaton
A titkosított szó: aebcahbcbgabba
```

Készíts programot, amely két funkcióra képes:

A. beolvas egy szót és kiírja titkosítva;

B. beolvas egy titkosított szót és kiírja a visszafejtését!

3. feladat: Alma (25 pont)

Egy almatermelő N fajta ($1 \le N \le 100$) almát termel, ismerjük, hogy melyik fajtából mennyit. Egy kereskedő M fajta ($1 \le M \le 100$) almát szeretne venni tőle, azt is tudjuk, hogy melyik fajtából mennyit.

Készíts programot, amely megadja, hogy a termelőnek melyik fajtából mennyi marad, valamint hogy a kereskedő melyik fajtából mennyit tud vásárolni!

Példa:

```
Bemenet:
Termelő fajtái száma: 3
1. fajta neve: jonagold
1. fajta mennyisége: 100
2. fajta neve: golden
2. fajta mennyisége: 30
3. fajta neve: idared
3. fajta mennyisége: 500
```

```
Kereskedő fajtái száma: 2
1. fajta neve: golden
1. fajta mennyisége: 50
2. fajta neve: starking
2. fajta mennyisége: 100
```

```
Kimenet:
Termelőnél marad:
jonagold 100
idared 500
```

```
Kereskedő vesz:
golden 30
```

Kilencedik-tizedik osztályosok

1. feladat: Állatkert (15 pont)

Egy állatkertben ismerjük a bejárható útvonalakat. A bejárat a 0-s sorszámú pont. Az egyes állatokat az 1 és N közötti sorszámukkal azonosítjuk, az utakat pedig két olyan állat sorszámával, amelyek ketrece között vezetnek. A látogatók szeretnének bizonyos állatokhoz a lehető legrövidebb útvonalon eljutni.

Készíts programot, amely az állatkerti utak ismeretében megadja, hogy a bejárattól melyik állathoz hányféleképpen lehet a lehető legrövidebb útvonalon eljutni!

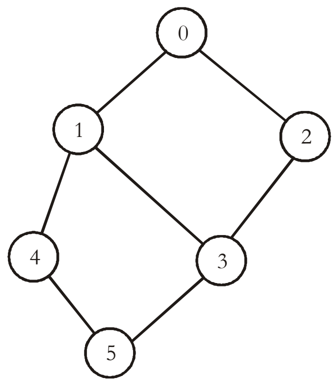
A bemenet első sorában az állatkert állatai száma ($1 \le N \le 40$) és az utak száma ($1 \le M \le 1000$) van. A következő M sor mindegyikében egy-egy út leírása található: azon két állat (vagy a bejárat) sorszáma, amelyek között az út vezet.

A kimenetre N sort kell írni! Az i-edik sorba a bejárattól az i sorszámú állathoz vezető legrövidebb utak száma kerüljön!

Példa:

```
Bemenet:
5 7
0 1
1 4
3 1
3 5
2 0
2 3
4 5
```

```
Kimenet:
1
1
2
1
3
```



2. feladat: Róka (15 pont)

Rókákat tenyésztünk: minden róka L évig él, a K, K+1, …,L éves rókák szaporodnak, minden ilyen évben 1-1 utódjuk születik. Tudjuk, hogy kezdetben hány 1 éves, 2-éves, … L éves rókánk van.

Készíts programot, amely megadja, hogy N év után hány rókánk lesz! Mivel ez a szám nagyon nagy is lehet, a számot MOD 1 000 000 kell kiírni!

A bemenet első sora az évek számát (1≤N≤100), a róka maximális korát (1≤L≤10) és az első évet, amikor szaporodhat (1≤K≤L) tartalmazza. A következő L sorból az i-edikben a kezdetben levő i-éves rókák száma van (0≤DB_i≤100).

A kimenet egyetlen sorába az N év után élő rókák számát kell írni (MOD 1 000 000)!

<u>Példa</u>:

```
Bemenet:          Kimenet:             magyarázat:
2 5 3                36                0. év: 2+3+4+5+6=20 róka
2                                      1. év: 15+2+3+4+5=29 róka
3                                      2. év: 12+15+2+3+4=36 róka
4
5
6
```

<u>3. feladat</u>: Gyöngy (15 pont)

Egy kör alakú nyakláncon N darab gyöngy van (többféle színűek). A nyakláncot az i-edik gyöngy után elvághatjuk (1≤i≤N).

Készíts programot, amely megadja, hogy hol vágjuk el a nyakláncot, hogy a K távolságon belül tőle balra levő piros, illetve jobbra levő zöld gyöngyök számának különbsége abszolút értékben a lehető legkisebb legyen!

A bemenet első sorában a gyöngyök száma (1≤N≤1 000 000) és a K szám (1≤K≤N/2) van. A következő sor N karaktert tartalmaz, közülük az i-edik az i-edik gyöngy színe. P betű jelöli a piros, Z betű a zöld gyöngyöket, a többi betű más színű gyöngyöt jelöl.

A kimenet első sorába annak a gyöngynek a sorszámát kell írni, amely után elvágva a nyakláncot, a K távolságon belül tőle balra levő piros, illetve jobbra levő zöld gyöngyök számának különbsége a lehető legkisebb lesz! Több megoldás esetén bármelyik megadható.

<u>Példa</u>:

```
Bemenet:                     Kimenet:
20 6                         14
ABPPZPPHHPPZZZZZPPPP
```

<u>4. feladat</u>: Raktár (15 pont)

Egy megye településeiről tudjuk, hogy bármely településről bármelyik másikra pontosan egy útvonalon lehet eljutni. Egy vállalat az összes településen nyit termelő üzemet. Tudjuk, hogy melyik településen van a raktár, ahova az egyes üzemek szállítanák az árut. Ismerjük, hogy melyik településen mennyi terméket gyárthatnak. Az egyes utakon egy nap maximum M mennyiségű termék szállítható.

Készíts programot, amely kiszámítja, hogy mekkorára kell építeni a raktárt, hogy a lehető legtöbb árut befogadja!

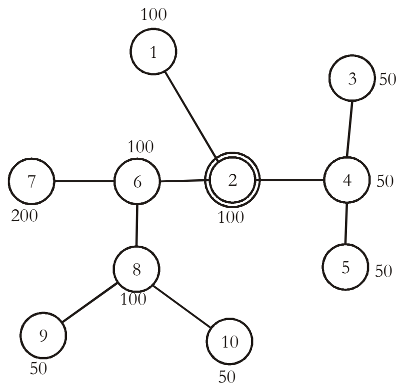
A bemenet első sorában a települések száma (1≤N≤1000), az utakon szállítható termékek maximális száma (1≤M≤100 000) és a raktáros település sorszáma (1≤R≤N) van. A második sor i. száma az i. településen gyártott áru mennyisége (1≤mennyiség≤1000). A következő N-1 sorban két település sorszáma van, amelyek között van út (1≤A≠B≤N).

A kimenet egyetlen sorába a maximális raktárméretet kell írni, ami az üzemekből szállított áruval egy nap megtölthető!

Példa:

```
Bemenet:
10 200 2
100 100 50 50 50 100 200 100 50 50
1 2
3 4
4 5
4 2
2 6
6 7
6 8
8 9
10 8
```

```
Kimenet:
550
```



5. feladat: Munka (15 pont)

Egy vállalkozó N munka ajánlatot kapott, de egyszerre csak egy munkát tud végezni. Minden ajánlatban szerepel, hogy a munkát mikor kellene elkezdeni, meddig tartana és a munka elvégzéséért mennyi fizetést kapna.

Készíts programot, amely kiszámítja, hogy a vállalkozó maximum mennyit kereshet!

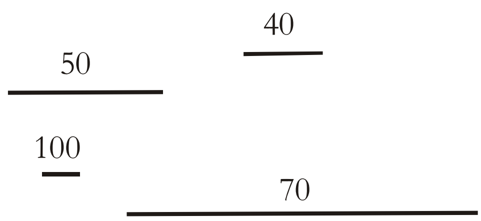
A bemenet első sorában a munkák száma van (1≤N≤1000). A következő N sor mindegyike az igényelt munka kezdő és befejező napjának sorszámát (1≤A≤B≤100 000), valamint pénzértékét (1≤érték≤10 000) tartalmazza.

A kimenet egyetlen sorába az elérhető legnagyobb jövedelem összegét kell írni!

Példa:

```
Bemenet:
4
30 40 40
0 20 50
5 10 100
15 60 70
```

```
Kimenet:
170
```



Tizenegyedik-tizenharmadik osztályosok

1. feladat: Szállítás (15 pont)

Az ország N városa között különböző teherbírású utak vannak. Két város között árut szeretnénk szállítani a lehető legnagyobb kapacitású teherautóval olyan útvonalon, ahol az autó tehersúlya nem nagyobb, mint az egyes utak teherbírása.

Készíts programot, amely adott A és B városra megadja, hogy maximum mekkora tehersúlyú teherautó közlekedhet közöttük és merre kell menni!

A bemenet első sorában a városok száma (1≤N≤100), a köztük levő utak száma (1≤M≤10 000), a kezdő és a cél város sorszáma (1≤A≠B≤N) van. A következő M sor mindegyikében egy-egy út leírása található: azon két város sorszáma (1≤sorszám≤N), amelyek között a kétirányú út vezet, valamint az út teherbírása (1≤tehebírás≤1000).

A kimenetre 2 sort kell írni! Az elsőbe a maximális tehersúly kerüljön, a másodikba pedig az oda vezető úton levő városok sorszáma, az útvonal sorrendjében (azaz az első sorszám biztosan A, az utolsó sorszám biztosan B legyen)! Több megoldás esetén bármelyik megadható.

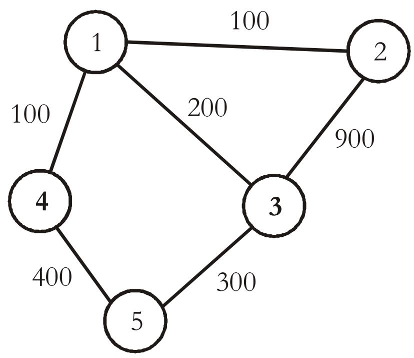
Példa:

Bemenet:

```
5 6 3 4
2 1 100
1 4 100
3 1 200
3 5 300
2 3 900
4 5 400
```

Kimenet:

```
300
3 5 4
```



2. feladat: Nyúl (15 pont)

Nyulakat tenyésztünk: minden nyúl L évig él, a K, K+1, …, L éves nyulak szaporodnak, minden ilyen évben az i éves nyulaknak R_i utódjuk születik. Tudjuk, hogy kezdetben hány 1 éves, 2-éves, … L éves nyulunk van.

Készíts programot, amely megadja, hogy N év után hány nyulunk lesz! Mivel ez a szám nagyon nagy is lehet, a számot MOD 1 000 000 kell kiírni!

A bemenet első sora az évek számát ($1 \le N \le 100$), a nyúl maximális korát ($1 \le L \le 10$) és az első évet tartalmazza, amikor szaporodhat ($1 \le K \le L$). A következő L sorból az i-edikben a kezdetben levő i éves nyulak száma van ($0 \le DB_i \le 100$), amit $i \ge K$ esetén az utódjai R_i száma követ.

A kimenet egyetlen sorába az N év után élő nyulak számát kell írni (MOD 1 000 000)!

Példa:

Bemenet:

```
2 5 3
2
3
4 1
5 2
6 1
```

Kimenet:

```
45
```

magyarázat:

```
0. év: 2+3+4+5+6=20 nyúl
1. év: 20+2+3+4+5=34 nyúl
2. év: 16+20+2+3+4=45 nyúl
```

3. feladat: Nyaklánc (15 pont)

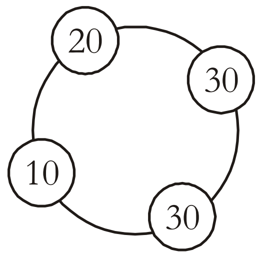
Egy körbefűzött nyakláncon N darab (N páros), különböző értékű gyöngy van. A nyakláncot az i-edik gyöngy után elvághatjuk. Az egyes gyöngyök értéke a vágás helyétől vett kisebb távolsággal megszorzódik (a két közvetlen szomszéd értéke egyszeres, az eggyel távolabbiak kétszeres, a még eggyel távolabbiak háromszoros értékűek lesznek, ... és így tovább).

Készíts programot, amely megadja, hogy hol vágjuk el a nyakláncot, hogy az összérték a lehető legnagyobb legyen!

A bemenet első sorában a gyöngyök száma van ($3 \le N \le 1\,000\,000$). A következő N sor mindegyike egy egész számot tartalmaz, közülük az i-edik az i-edik gyöngy értékét ($1 \le érték_i \le 100$).

A kimenet első sorába annak a gyöngynek a sorszámát kell írni, amely után elvágva a nyakláncot, a gyöngyök összértéke a lehető legnagyobb lesz! Több megoldás esetén bármelyik megadható.

Példa:




```
Bemenet:          Kimenet:
4                 3
30
30
10
20
```

4. feladat: Üzletek (15 pont)

Egy megye településeiről tudjuk, hogy bármely településről bármelyik másikra pontosan egy útvonalon lehet eljutni. Egy üzlethálózat minél több településen szeretne üzletet nyitni. Tudjuk, hogy melyik településen van a raktár, ahonnan az egyes üzletekbe szállítaná az árut. Ismerjük, hogy melyik településen mekkora hasznot fog hozni az üzlet, valamint hogy melyik út használatáért mekkora úthasználati díjat kell fizetni (ez nem függ attól, hogy hány településről viszik ezen az úton az árut). Az összhaszon az üzletekben termelt haszonból levonva az úthasználati díjakat.

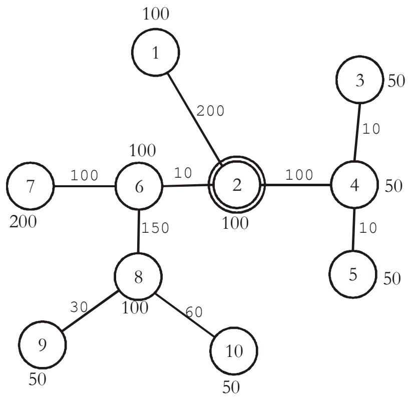
Készíts programot, amely kiszámítja, hogy mekkora az elérhető legnagyobb haszon és ehhez mely településeken kell üzletet nyitni!

A bemenet első sorában a települések száma (1≤N≤10 000) és a raktáros település sorszáma (1≤R≤N) van. A második sor i-edik száma az i. településen elérhető haszon. A következő N-1 sorban két település sorszáma, amelyek között kétirányú út van, és az érte fizetendő úthasználati díj található.

A kimenet első sorába az elérhető legnagyobb hasznot kell írni! A második sorba a megnyitandó üzletek M számát kell írni, a harmadikba M sorszámot: azon települések sorszámát, ahol üzletet nyitunk! A számok sorrendje közömbös. Több megoldás esetén bármelyik megadható.

Példa:

```
Bemenet:                                Kimenet:
10 2                                    320
100 100 50 50 50 100 200 100 50 50      6
1 2 200                                 2 3 4 5 6 7
3 4 10
4 5 10
4 2 100
2 6 10
6 7 100
6 8 150
8 9 30
10 8 60
```



5. feladat: Párok (15 pont)

Az iskola szalagavató bálján a 11. osztályosok tűzik ki a szalagokat a 12. osztályosokra. Mindenkiről tudjuk, hogy a másik osztályból kit ismer. A két osztály felsorakozik egymással szemben, de mindenki csak valamely ismerősének akarja kitűzni a szalagot. Feltesszük, hogy a két osztály azonos létszámú (mindkettőben N tanuló van).

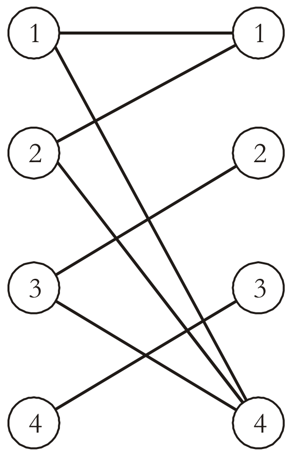
Készíts programot, amely megadja, hogy egyszerre maximum hány 11. osztályos diák tűzheti ki úgy a szalagot valamely 12. osztályos ismerőse ruhájára, hogy a hozzálépése során egyik diák sem keresztezheti semelyik másik diák útját!

A bemenet első sorában a tanulók száma (`1≤N≤500`) és a baráti párok száma (`1≤M≤20 000`) van. A következő M sor mindegyike két egész számot tartalmaz: a baráti párokat.

A kimenet első sorába azon 11.-es tanulók maximális M számát kell írni, akik útjuk keresztezése nélkül odaléphetnek valamely 12.-es ismerősükhöz a szalagot kitűzni! A következő M sorban egy-egy pár 11.-es és 12.-es tagjának sorszáma szerepeljen, sorszám szerint növekvő sorrendben!

Példa:

```
Bemenet:        Kimenet:
4 7             3
1 1             1 1
1 4             3 2
2 1             4 3
3 2
4 3
2 4
3 4
```



2012. A verseny végeredménye

I. korcsoport

1	Alexy Marcell	Juhász Gyula Általános Iskola, Vác
2	Zarándy Álmos	Fazekas Mihály Gimnázium, Budapest
	Baran Zsuzsanna	Fazekas Mihály Gimnázium, Debrecen
4	Almási Nóra	Fazekas Mihály Gimnázium, Debrecen
5	Nagy Máté	Fazekas Mihály Gimnázium, Debrecen
	Radnai László	Veres Péter Gimnázium, Budapest
6	Pásztor Szabolcs	Fazekas Mihály Gimnázium, Debrecen
7	Szeder Patrik	Széchenyi István Gimnázium, Dunaújváros
	Pecsenye Samu	Fazekas Mihály Gimnázium, Debrecen
8	Csesznák Tamás	Református Gimnázium, Szentendre
9	Vankó Dániel	Bárdos László Gimnázium, Tatabánya
10	Augusztin András	Központi Általános Iskola és Diákotthon, Salgótarján

II. korcsoport

1	Székely Szilveszter	Neumann János Középiskola és Kollégium, Eger
2	Somogyvári Kristóf	Ságvári Endre Gimnázium, Szeged
3	Weisz Ambrus	Fazekas Mihály Gimnázium, Budapest
4	Leitereg Miklós	Veres Péter Gimnázium, Budapest
5	Végvári Zalán	Fazekas Mihály Gimnázium, Debrecen
	Kolumbán Antal-György	Tamási Áron Elméleti Líceum, Székelyudvarhely
7	Nagy Vendel	Fazekas Mihály Gimnázium, Debrecen
8	Virág Fausztin Asztrik	Földes Ferenc Gimnázium, Miskolc
	Kocsis Dávid	Gábor Dénes Gimn. és Müszaki Szakközépiskola, Szeged
10	Kurkó Mihály Zsolt	Márton Áron Gimnázium, Csíkszereda
	Erdős Márton	Batthyány Lajos Gimnázium, Nagykanizsa

III. korcsoport

1	Szenczi Zoltán	Fazekas Mihály Gimnázium, Budapest
2	Marussy Kristóf	Szent István Gimnázium, Budapest
3	Weisz Gellért	Fazekas Mihály Gimnázium, Budapest
4	Havasi Márton	Fazekas Mihály Gimnázium, Budapest
5	Kovács Gábor Ferenc	Árpád Gimnázium, Tatabánya
6	Adrián Patrik	Baross Gábor Szakképző Intézet, Debrecen
7	Mezei Balázs Ferenc	Fazekas Mihály Gimnázium, Budapest
8	Mihálykó András	Lovassy László Gimnázium, Veszprém
9	Szabó Attila	Leőwey Klára Gimnázium, Pécs
10	Bogye Balázs	Balassi Bálint Nyolcévfolyamos Gimnázium, Budapest

11	Ódor Gergely	Fazekas Mihály Gimnázium, Budapest
12	Radnai Balázs	Fazekas Mihály Gimnázium, Budapest

2013. Első forduló

Ötödik-nyolcadik osztályosok

Számítógép nélküli feladatok

1. feladat: Karesz a robot (44 pont)

Karesz egy „utcagyerek”, aki egy négyzetrácsos úthálózat utcasarkain lépdelve mozog. Egyes utcasarkokon kavicsok találhatók, amiket Karesz tetszőleges számban felvehet, majd újra letehet. Karesz, mint algoritmus végrehajtó, az alábbi nyelvet érti:

Karesz tevékenységei

- `Fordulj jobbra`
- `Fordulj balra`
- `Lépj`
- `Vegyél fel egy kavicsot`
- `Tegyél le egy kavicsot`

Amit Kareszről kérdezhetünk

- `északra néz`
- `délre néz`
- `keletre néz`
- `nyugatra néz`
- `van itt kavics`

Kezdetben Karesz a (10,10) koordinátájú ponton áll és felfelé néz (azaz ha lépne egyet, akkor a (10,11) koordinátájú pontra lépne. Kezdetben a (10,10) koordinátájú ponton X darab kavics van, a (11,10) koordinátájún pedig Y darab kavics van, a többi ponton nincs kavics.

A. Melyik mezőn fog állni Karesz az alábbi program végrehajtása után, ha X=1, Y=2?

B. Melyik mezőn fog állni Karesz az alábbi program végrehajtása után, ha X=2, Y=1?

C. Mitől és hogyan függ tetszőleges X és Y esetén, hogy Karesz az alábbi program végrehajtása után melyik mezőn fog állni?

D. Az alábbi program végrehajtása után tetszőleges X és Y esetén melyik mezőkön lesz kavics és mennyi?

```
Karesz:
  Ciklus amíg van itt kavics
    Vegyél fel egy kavicsot
    Lépj
    Tegyél le egy kavicsot
    Fordulj jobbra
    Lépj
    Fordulj jobbra
    Lépj
  Ciklus vége
Eljárás vége.
```

2. feladat: Mit csinál (44 pont)

Az alábbi algoritmus az N elemű X vektor alapján számolja ki M számot és az Y vektor értékeit.

```
Valami(N,X,M,Y):
  M:=1; K:=X(1); Y(M):=1
  Ciklus i=2-től N-ig
    Ha X(i)>K akkor M:=1; K:=X(i); Y(M):=i
    különben Ha X(i)=K akkor M:=M+1; Y(M):=i
  Ciklus vége
Eljárás vége.
```

A. Mi lesz M és az Y vektor értéke, ha N=3, X =(-5,3,2)?

B. Mi lesz M és az Y vektor értéke, ha N=3, X =(-5,3,3)?

C. Mi lesz M és az Y vektor értéke, ha N=3, X =(3,3,3)?

D. Fogalmazd meg általánosan, hogy milyen feltétel esetén lesz az eljárás végén M=1!

E. Fogalmazd meg általánosan, hogy milyen feltétel esetén lesz az eljárás végén M=N!

F. Fogalmazd meg általánosan, hogy hogyan függ adott i-re M és Y értéke a bemenettől!

G. Fogalmazd meg általánosan, hogy hogyan függ az eljárás végére érve M és Y értéke a bemenettől!

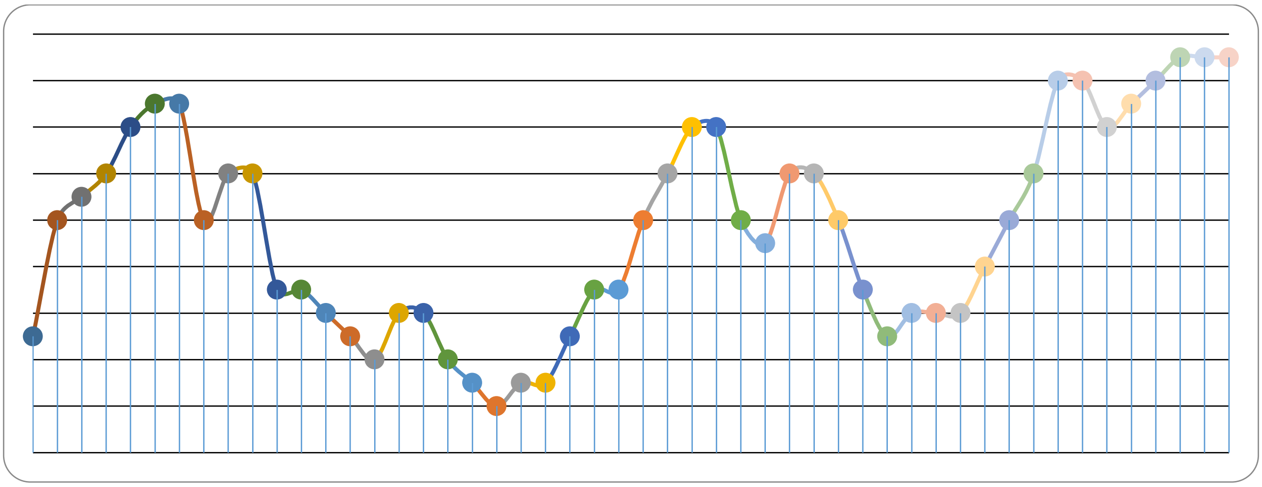
3. feladat: Hegyek-völgyek (52 pont)

Az alábbi algoritmus N adatot kap bemenetként (`N>2, X(1)≤X(2), X(N)≤X(N-1)`) az X vektorban, amelyekből több értéket számol ki:

```
Valami(N,X,???):
  H:=0; Ha X(1)<X(N) akkor L:=1 különben L:=N
  Ciklus i=2-től N-1-ig
    Ha X(i)<X(L) akkor L:=i
    Ha X(i)≥X(i-1) és X(i)>X(i+1) akkor VK:=i; VH:=0
    VH:=VH+1
    Ha X(i)>X(i-1) és X(i)≥X(i+1)
       akkor VV:=i; Ha VH>H akkor H:=VH; K:=VK; V:=VV
  Ciklus vége
Eljárás vége.
```

A. Melyik változók lehetnek az eljárás kimenő paraméterei (az eljárásfejben mit kell a kérdőjelek helyére írni – a kérdőjelek száma nem biztos, hogy azonos a kimenő paraméterek számával)?

B. Melyik kimenő paraméternek mi lesz az értéke az alábbi 50 elemű X vektor esetén?



C. Fogalmazd meg általánosan az egyes kimenő paraméterek szerepét!

D. Mi a szerepe azoknak a lokális változóknak (az i változó kivételével), amelyek nem kimenő paraméterek?

E. Előfordulhat bizonyos X vektorokra, hogy egyes kimenő változók nem kapnak értéket. Milyen X vektorra lehetséges ez?

Számítógépes feladat – VÁLASZTHATÓ

4. feladat: Lakások (60 pont)

Egy ingatlanforgalmazó cég tárolja az eladó lakások alapterületét és árát. Írj programot, amely megadja

- a legdrágább lakás sorszámát (ha több megoldás van, akkor közülük a legkisebb sorszámút);
- a 100 négyzetméternél nagyobbak közül a 40 millió forintnál olcsóbbak számát;
- hányféle alapterületű lakás van!

A bemenet első sorában a lakások száma (≥1), alatta soronként egy-egy lakás alapterülete (négyzetméterben) és ára van (millió forintban, egész számok). A kimenetre kell kiírni a fenti kérdésekre, feladatokra adott válaszokat.

Minta:

Bemenet (billentyűzet)		Kimenet (képernyő)	
#	Sortartalom [*magyarázat*]	#	Sortartalom [*magyarázat*]
1.	6 [*1≤lakások száma≤100*]	1.	4 [*az 1. részfeladat válasza*]
2.	42 15 [*az 1. lakás alapterülete és ára*]	2.	2 [*a 2. részfeladat válasza*]
3.	110 20 [*a 2. lakás alapterülete és ára*]	3.	4 [*a 3. részfeladat válasza*]
4.	125 160 [*a 3. lakás alapterülete és ára*]		
5.	166 180 [*a 4. lakás alapterülete és ára*]		
6.	42 10 [*az 5, lakás alapterülete és ára*]		
7.	110 39 [*a 6. lakás alapterülete és ára*]		

Számítógép nélküli feladat – VÁLASZTHATÓ

4. feladat: Lakások (60 pont)

Egy ingatlanforgalmazó cég tárolja az eladó lakások alapterületét és árát. Programot írtunk, amely megadja:

A. a legkisebb lakás sorszámát (ha több megoldás van, akkor közülük a legnagyobb sorszámút);

B. a 100 négyzetméternél kisebbek átlagárát;

C. a 100 millió forintnál drágább lakások számát és sorszámait.

Az alábbi eljárásban N a lakások száma, a T vektor a lakások alapterületét, az A vektor pedig az árát tartalmazza:

```
Lakás(N,T,A,M,B,L,X):
  M:=1; D:=0; S:=0; L:=0
  Ciklus i=1-től N-ig
    Ha ☐☐☐☐☐☐☐☐☐ akkor M:=☐☐☐☐☐
    Ha ☐☐☐☐☐☐☐☐☐ akkor D:=☐☐☐☐☐; S:=☐☐☐☐☐
    Ha ☐☐☐☐☐☐☐☐☐ akkor L:=☐☐☐☐☐; X(L):=☐☐☐☐☐
  Ciklus vége
  Ha ☐☐☐☐☐☐☐☐☐ akkor B:=☐☐☐☐☐ különben B:=0
Eljárás vége.
```

Mit kell írni a téglalapok helyére, hogy a fenti feladatok helyes megoldását kapjuk? (Új sorok nem írhatók, új változó nem vezethető be.)

Kilencedik-tizedik osztályosok

1. feladat: Karesz a robot (44 pont)

Karesz egy „utcagyerek”, aki egy négyzetrácsos úthálózat utcasarkain lépdelve mozog. Egyes utcasarkokon kavicsok találhatók, amiket Karesz tetszőleges számban felvehet, majd újra letehet. Karesz, mint algoritmus végrehajtó, az alábbi nyelvet érti:

Karesz tevékenységei

- `Fordulj jobbra`
- `Fordulj balra`
- `Lépj`
- `Vegyél fel egy kavicsot`
- `Tegyél le egy kavicsot`

Amit Kareszről kérdezhetünk

- `északra néz`
- `délre néz`
- `keletre néz`
- `nyugatra néz`
- `van itt kavics`

Kezdetben Karesz a (10,10) koordinátájú ponton áll és felfelé néz (azaz ha lépne egyet, akkor a (10,11) koordinátájú pontra lépne. Kezdetben a (10,10) koordinátájú ponton X darab kavics van, a (11,10) koordinátájún pedig Y darab kavics van, a többi ponton nincs kavics.

A. Melyik mezőn fog állni Karesz az alábbi program végrehajtása után, ha X=1, Y=2?

B. Melyik mezőn fog állni Karesz az alábbi program végrehajtása után, ha X=2, Y=1?

C. Mitől és hogyan függ tetszőleges X és Y esetén, hogy Karesz az alábbi program végrehajtása után melyik mezőn fog állni?

D. Az alábbi program végrehajtása után tetszőleges X és Y esetén melyik mezőkön lesz kavics és mennyi?

```
Karesz:
  Ciklus amíg van itt kavics
    Vegyél fel egy kavicsot
    Lépj
    Tegyél le egy kavicsot
    Fordulj jobbra
    Lépj
    Fordulj jobbra
    Lépj
    Fordulj jobbra
    Fordulj jobbra
    Ha van itt kavics akkor Vegyél fel egy kavicsot
                            Lépj
                            Tegyél le egy kavicsot
                            Fordulj balra
                            Lépj
                            Fordulj balra
                            Lépj
                            Fordulj balra
                            Fordulj balra
    Elágazás vége
  Ciklus vége
Eljárás vége.
```

2. feladat: Mit csinál (54 pont)

Az alábbi algoritmus az N elemű, pozitív számokat tartalmazó X vektor alapján számolja ki M számot és az Y vektor értékeit.

```
Valami(N,X,M,Y):
  M:=0; K:=-2
  Ciklus i=1-től N-ig
    Ha X(i) mod 2=0 akkor
       Ha X(i)>K akkor M:=1; K:=X(i); Y(M):=i
       különben Ha X(i)=K akkor M:=M+1; Y(M):=i
    Elágazás vége
  Ciklus vége
Eljárás vége.
```

A. Mi lesz M és az Y vektor értéke, ha N=3, X =(5,3,2)?

B. Mi lesz M és az Y vektor értéke, ha N=3, X =(5,4,4)?

C. Mi lesz M és az Y vektor értéke, ha N=3, X =(6,6,6)?

D. Fogalmazd meg általánosan, hogy milyen feltétel esetén lesz az eljárás végén M=0!

E. Fogalmazd meg általánosan, hogy milyen feltétel esetén lesz az eljárás végén M=1!

F. Fogalmazd meg általánosan, hogy milyen feltétel esetén lesz az eljárás végén M=N!

G. Fogalmazd meg általánosan, hogy hogyan függ adott i-re M és Y értéke a bemenettől!

H. Fogalmazd meg általánosan, hogy hogyan függ az eljárás végére érve M és Y értéke a bemenettől!

I. Mi a szerepe a K változónak?

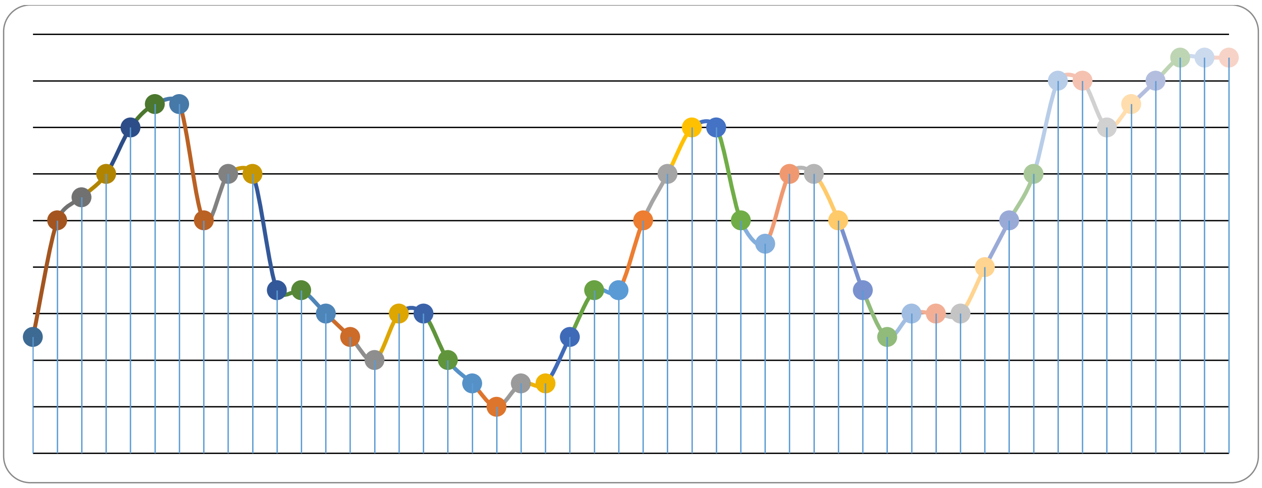
3. feladat: Hegyek-völgyek (52 pont)

Az alábbi algoritmus N adatot kap bemenetként (`N>2, X(1)≤X(2), X(N)≤X(N-1)`) az X vektorban (`X(i)>0`), amelyekből több értéket számol ki:

```
Valami(N,X,???):
  H:=0; M:=0; VH:=0
  Ciklus i=2-től N-1-ig
    Ha X(i)≥X(i-1) és X(i)>X(i+1) akkor VK:=i; VH:=0
    VH:=VH+1
    Ha X(i)>X(i-1) és X(i)≥X(i+1)
       akkor VV:=i; Ha VH>H akkor H:=VH; K:=VK; V:=VV
             Ha X(VK)>X(VV) akkor Ha X(VV)>M akkor M:=X(VV)
             Ha X(VK)≤X(VV) akkor Ha X(VK)>M akkor M:=X(VK)
  Ciklus vége
Eljárás vége.
```

A. Melyik változók lehetnek az eljárás kimenő paraméterei (az eljárásfejben mit kell a kérdőjelek helyére írni – a kérdőjelek száma nem biztos, hogy azonos a kimenő paraméterek számával)?

B. Melyik kimenő paraméternek mi lesz az értéke az alábbi 50 elemű X vektor esetén?



C. Fogalmazd meg általánosan az egyes kimenő paraméterek szerepét!

D. Mi a szerepe azoknak a lokális változóknak (az i változó kivételével), amelyek nem kimenő paraméterek?

E. Előfordulhat bizonyos X vektorokra, hogy egyes kimenő változók nem kapnak értéket. Milyen X vektorra lehetséges ez?

<u>4. feladat</u>: Raktár (50 pont)

Egy áruházlánc két raktárból látja el a hozzá tartozó áruházakat egy adott termékkel. Az A raktárban M, a B raktárban N darab termék van, az áruházak száma M+N. Minden áruházba egy darab terméket kell szállítani, és ismerjük minden áruházra az A, illetve a B raktárból történő szállítás költségét (`A(i)`, illetve `B(j)`).

A. Add meg az alábbi adatokra, hogy melyik raktárból kell szállítani az egyes áruházakba, hogy a szállítás összköltsége minimális legyen! Add meg a minimális költséget is!

N=3, M=3

A=1 2 3 4 3 2

B=1 3 2 7 3 1

B. Definiáld rekurzívan azt a táblázatot (`Költség(I,J)`), amely annak minimális költségét tartalmazza, hogy az első `I+J` áruházba kell szállítani árut, `I` darabot az első, `J` darabot a második raktárból!

Tizenegyedik-tizenharmadik osztályosok

<u>1. feladat</u>: Karesz a robot (40 pont)

Karesz egy „utcagyerek”, aki egy négyzetrácsos úthálózat utcasarkain lépdelve mozog. Egyes utcasarkokon kavicsok találhatók (egy helyen csak egy), amiket Karesz tetszőleges számban felvehet, majd újra letehet. Karesz kezében kezdetben Z darab kavics van. Karesz, mint algoritmus végrehajtó, az alábbi nyelvet érti:

Karesz tevékenységei	Amit Kareszről kérdezhetünk
Fordulj jobbra	északra néz
Fordulj balra	délre néz
Lépj	keletre néz
Vegyél fel egy kavicsot	nyugatra néz
Tegyél le egy kavicsot	van itt kavics

Kezdetben Karesz a (10,10) koordinátájú ponton áll és felfelé néz (azaz ha lépne egyet, akkor a (10,11) koordinátájú pontra lépne. Kezdetben a (10,x) koordinátájú ponton 1 kavics van, a (10,y) koordinátájún szintén 1 kavics van (10<x<y, x+y páros szám), a többi ponton nincs kavics. Ha Karesz nem tud letenni kavicsot, mert nincs a kezében, akkor hibát jelez és leáll.

A. Karesz az alábbi program végrehajtása után melyik mezőn fog állni, ha X=15, Y=17, Z=0?

B. Karesz az alábbi program végrehajtása után melyik mezőn fog állni, ha X=15, Y=19, Z=0?

C. Karesz az alábbi program végrehajtása után melyik mezőn fog állni tetszőleges X és Y esetén?

D. Az alábbi program végrehajtása után melyik mezőkön lesz kavics és mennyi, illetve hány kavics lesz Karesz kezében tetszőleges X, Y és Z esetén?

E. Milyen X,Y,Z esetén működne hibamentesen a program, ha kivennénk a két „`Vegyél fel egy kavicsot`” utasítást?

F. Mi változna az E pont szerint módosított algoritmus hibamentes működése esetén?

```
Karesz:
  Ciklus amíg nem van itt kavics
    Lépj
  Ciklus vége
  Vegyél fel egy kavicsot
  Lépj
  Ciklus amíg nem van itt kavics
    Tegyél le egy kavicsot
    Lépj
    Ciklus amíg nem van itt kavics
      Lépj
    Ciklus vége
    Fordulj jobbra
    Fordulj jobbra
    Vegyél fel egy kavicsot
    Lépj
  Ciklus vége
Eljárás vége.
```

2. feladat: Mit csinál (46 pont)

Az alábbi algoritmus az N elemű, 0 és 100 közötti pozitív számokat tartalmazó X vektor alapján számolja ki a P, Q számokat és az Y, Z vektorok értékeit.

```
Valami(N,X,P,Y,Q,Z):
  P:=0; Q:=0; K:=-2; L:=101
  Ciklus i=1-től N-ig
    Ha X(i) mod 2=0 akkor
       Ha X(i)>K akkor P:=1; K:=X(i); Y(P):=i
       különben Ha X(i)=K akkor P:=P+1; Y(P):=i
    különben Ha X(i)<L akkor Q:=1; L:=X(i); Z(Q):=i
             különben Ha X(i)=L akkor Q:=Q+1; Z(Q):=i
    Elágazás vége
  Ciklus vége
Eljárás vége.
```

A. Mi lesz a P, Q és az Y, Z vektorok értéke, ha N=3, X =(3,5,2)?

B. Mi lesz a P, Q és az Y, Z vektorok értéke, ha N=4, X =(5,4,5,4)?

C. Milyen feltétel esetén lesz az eljárás végén P=0?

D. Milyen feltétel esetén lesz az eljárás végén Q=0?

E. Fogalmazd meg általánosan, hogy milyen feltétel esetén lesz az eljárás végén P=1!

F. Fogalmazd meg általánosan, hogy milyen feltétel esetén lesz az eljárás végén Q=N!

G. Fogalmazd meg általánosan, hogy hogyan függ adott i-re P, Q és Y, Z értéke a bemenettől!

H. Fogalmazd meg általánosan, hogy hogyan függ az eljárás végére érve P, Q és Y, Z értéke a bemenettől!

I. Mi a szerepe a K és az L változóknak?

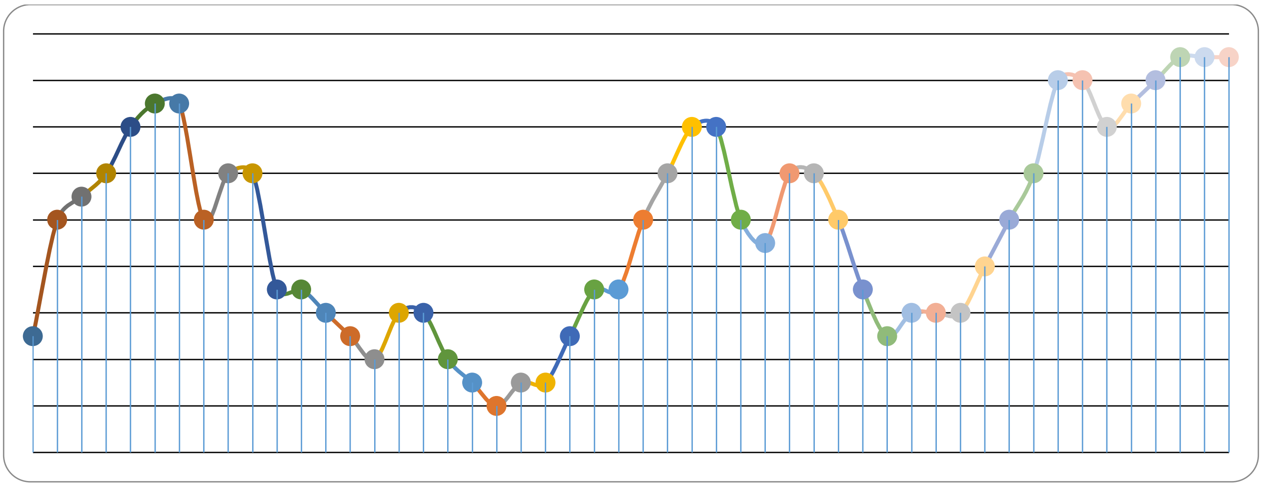
3. feladat: Hegyek-völgyek (44 pont)

Az alábbi algoritmus N adatot kap bemenetként (`N>2, X(1)≤X(2), X(N)≤X(N-1)`) az X vektorban, amelyekből több értéket számol ki:

```
Valami(N,X,???):
  H:=0; M:=0 ; VH:=0; VK:=1; VA:=1
  Ciklus i=2-től N-1-ig
    Ha X(i)≥X(i-1) és X(i)>X(i+1) akkor VK:=i; VH:=0
    VH:=VH+1
    Ha X(i)<X(i-1) és X(i)<X(i+1) akkor VA:=i
    Ha X(i)>X(i-1) és X(i)≥X(i+1)
       akkor VV:=i; Ha VH>H akkor H:=VH; K:=VK; V:=VV
             Ha X(VK)>X(VV) akkor Ha X(VV)-X(VA)>M
                                  akkor M:=X(VV)-X(VA)
             Ha X(VK)≤X(VV) akkor Ha X(VK)-X(VA)>M
                                  akkor M:=X(VK)-X(VA)
  Ciklus vége
Eljárás vége.
```

A. Melyik változók lehetnek az eljárás kimenő paraméterei (az eljárásfejben mit kell a kérdőjelek helyére írni – a kérdőjelek száma nem biztos, hogy azonos a kimenő paraméterek számával)?

B. Melyik kimenő paraméternek mi lesz az értéke az alábbi 50 elemű X vektor esetén?



C. Fogalmazd meg általánosan az egyes kimenő paraméterek szerepét!

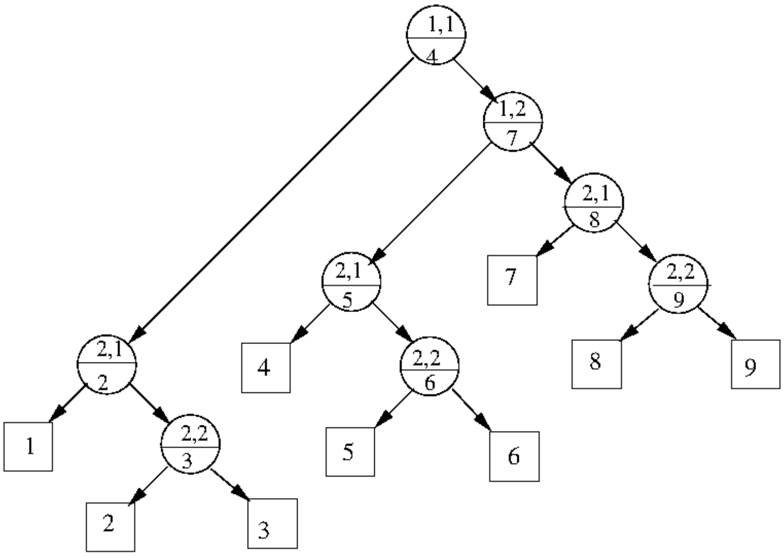
D. Mi a szerepe azoknak a lokális változóknak (az `i` változó kivételével), amelyek nem kimenő paraméterek?

E. Előfordulhat bizonyos X vektorokra, hogy egyes kimenő változók nem kapnak értéket. Milyen X vektorra lehetséges ez?

F. Mi lenne a szerepe annak, ha a feldolgozás előtt `X(i)` helyébe mindenhol `X(i) DIV 10`-et tennénk?

4. feladat: Anyagvizsgálat (35 pont)

Különleges anyag szilárdságát kell megállapítani, amihez anyagmintákat használhatunk. Az anyagmintát úgy használják, hogy adott erővel ráütnek az anyagmintára, és ha az összetörik, akkor az anyag szilárdsága kisebb, ha nem törik össze, akkor nagyobb vagy egyenlő a ráütő erővel. Ha nem törött össze az anyagminta, akkor újból felhasználható, de csak korlátozott alkalommal.



Például, ha egyféle mintából két darabunk van, mindegyik kétszer használható, akkor 8 a legnagyobb szilárdsági érték, amit a két anyagmintával meg tudunk állapítani.

Egy vizsgálati stratégiát fejez ki az ábra. A fa leveleiben a lehetséges szilárdsági értékek vannak balról jobbra növekvően. A fa belső pontjában lévő i,j számpár azt jelenti, hogy az `i.` mintát `j.` alkalommal használjuk, a harmadik szám pedig azt adja meg, hogy mekkora erővel ütünk a mintára. Ha a minta összetörik, akkor a bal részfával, ha nem, akkor a jobb részfával folytatódik a tesztelés.

Kétfajta anyagmintánk van, az egyik A, a másik B alkalommal használható, ha nem törik össze. Az első típusból U, a másikból V darab van.

A. A=1, B=3, U=2, V=1, mekkora a legnagyobb mérhető szilárdság?

B. A=1, B=3, U=2, V=1, első teszteléskor mekkora erővel kell ütni a mintára?

C. Ha csak egyfajta anyagmintánk van, adott A és U esetén add meg képlettel, hogy mekkora a legnagyobb szilárdság, ami mérhető!

D. Adott A ,B és U, V esetén add meg képlettel, hogy mekkora a legnagyobb szilárdság, ami mérhető!

E. Adott A, B és U, V esetén add meg képlettel, hogy mekkora erővel kell először ütni a mintára!

5. feladat: Munkavállalás (35 pont)

Egy vállalkozó 1 napos munkákat vállal. Ismerjük mindegyik munka (M darab) határidejét (1≤`H(i)`≤N). Egy nap csak 1 munkát végezhet. Add meg, hogy az M munka közül N napra maximum hány munkát vállalhat el és mely napokra melyikeket!

A feladat megoldására két algoritmust készítettünk.

1. megoldás:

```
Kiválogatás(N,M,H,Db,Nap):
  DB:=0; Nap():=(0,…,0)
  Ciklus i=1-től M-ig
    Ciklus amíg H(i)>0 és Nap(H(i))>0
      H(i):=H(i)-1                                  {*}
    Ciklus vége
    Ha H(i)>0 akkor Db:=Db+1; Nap(H(i)):=i
  Ciklus vége
Eljárás vége.
```

2. megoldás:

```
Kiválogatás(N,M,H,Db,Nap):
  S():=(1,2,…,M)
  Ciklus i=1-től M-1-ig
    min:=i
    Ciklus j=i+1-től M-ig
      Ha H(j)<H(min) akkor min:=j                   {**}
    Ciklus vége
    Csere(H(i),H(min)); Csere(S(i),S(min))
  Ciklus vége
  DB:=1; Nap(Db):=S(1)
  Ciklus i=2-től N-ig
    Ha Db<H(i) akkor Db:=Db+1; Nap(Db):=S(i)    {***}
  Ciklus vége
Eljárás vége.
```

A. Fogalmazd meg, hogy mi és hol szerepel a Nap tömbben az első, illetve a második változat esetén?

B. Az elvállalható munkákat a határidejükhöz képest mikorra osztja be a két változat?

C. Hogyan függ a *-gal jelölt sorok végrehajtási száma N-től, illetve M-től? (A kérdésre ilyen jellegű válaszok adhatók: N, N*M, N^2+M^3, …)

D. Az első algoritmus *-os sorhoz tartozó végrehajtási száma milyen N és M értékekre kisebb, mint a második algoritmus *-os sorokhoz tartozó végrehajtási száma?

2013. Második forduló

Ötödik-nyolcadik osztályosok

1. feladat: Vizsga (50 pont)

Egy vizsgabizottságban egy nap feljegyezték, hogy az egyes vizsgázók mikor vizsgáztak (egyszerre egy vizsgázó lehet), mindenkiről 4 adatot tudunk: `kezdőóra, kezdőperc, végóra, végperc.` A vizsgázók adatait idő szerinti sorrendben kapjuk.

Készíts programot, amely beolvassa a vizsgázók számát ($1\leq N\leq 100$), az egyes vizsgázók vizsgájának kezdetét és végét, majd megadja:

A. annak a 60 perces időszaknak a kezdetét, amikor a legtöbb vizsgázó végzett a vizsgájával (közülük az első pontosan ebben a percben végezzen);

B. a leghosszabb vizsgaszünet hosszát – óra, perc (amikor 2 vizsgázó között senki sem volt a vizsgabizottságnál);

C. a leghosszabb időtartamot, amikor a vizsgabizottság nem tarthatott szünetet!

Példa:

Bemenet:	Kimenet:	
N=6	10 10	(10 óra 10 perctől)
8 20 8 30	1 0	(1 óra, 0 perc)
8 50 9 0	1 5	(1 óra, 5 perc)
9 50 10 10		
10 10 10 30		
10 30 10 55		
11 55 12 10		

2. feladat: Számok (50 pont)

Számítógépen csak véges sok számjeggyel leírható törteket tudunk ábrázolni. Minden más X számra tudunk mondani egy alsó és egy felső határt, amire $AH\leq X\leq FH$. AH és FH egy-egy tört, aminek ismerjük a számlálóját és a nevezőjét, egyszerűsítve.

Így a π szám például a $\left[\frac{22}{7},\frac{223}{71}\right]$ intervallummal ábrázolható.

Írj programot, amely két pozitív szám ($X, Y\geq 0$, az eredmények alsó határa is ≥ 0) alsó és felső határa ismeretében megadja X+Y, X-Y, X*Y alsó és felső határát!

Példa:

Bemenet	Kimenet
X: $\left[\frac{7}{3},\frac{3}{1}\right]$	X+Y: $\left[\frac{23}{6},\frac{19}{4}\right]$
Y: $\left[\frac{3}{2},\frac{7}{4}\right]$	X-Y: $\left[\frac{7}{12},\frac{3}{2}\right]$
	X*Y: $\left[\frac{7}{2},\frac{21}{4}\right]$

3. feladat: Autók (50 pont)

Egy utat középen egy gyalogosátkelő két szakaszra oszt, a zebrához közlekedési lámpát helyeztek. Az útszakaszokat négyzetes cellákra osztjuk. N cella van a lámpa előtt, 1 cella a zebra, újabb N cella van a lámpa mögött. A mozgás szabályai:

- egy autó egy időegység alatt egy cellával mozdulhat el;

  Pl.:

X		X			X		X		→		X		X			X		X

- egy útszakaszon két autó között mindig kell lenni legalább 1 üres cellának (akkor is, ha sűrűbben érkeznének);

  Pl.:

X		X			X		X	

- a közlekedési lámpa minden P időtartam végén levő U időegységben piros, a többiben zöld; piros lámpaállásnál autó nem léphet a zebrára.

  Pl.:

X			X		X		X		→		X		X			X		X

Készíts programot, amely beolvassa a 2 útszakasz hosszát (`1≤N≤1000`), a bejövő autók számát balról (`1≤B≤100`), a bejövő autók belépési idejét (1 és 1000 közötti számok, növekvő sorrendben), valamint a P és az U időtartamot (`U<P`), majd megadja, hogy az egyes autók mikor jutnak ki az útszakasz végén! (Ha nem lenne lámpa, akkor a kilépési idejük pontosan 2*N+1-gyel lenne több, mint a belépési idejük.)

```
Bemenet:                    Kimenet:
N=4                         Kilép: 13, 15, 20
B=3
Belép=3, 6, 9
P=7, U=3
```

Idő=3	X								
Idő=4		X							
Idő=5			X						
Idő=6	X			X					
Idő=7		X		X					
Idő=8			X		X				
Idő=9	X			X		X			
Idő=10		X			X		X		
Idő=11			X			X		X	
Idő=12				X			X		X
Idő=13				X				X	
Idő=14				X					X
Idő=15					X				

Kilencedik-tizedik osztályosok

1. feladat: Iskola (40 pont)

Egy iskola tanárairól tudjuk, hogy mikor milyen órát tartanak. A tanárokat, a tantárgyakat, a hét napjait, a napokon belüli órákat sorszámukkal azonosítjuk. Készíts programot, amely megadja:

A. minden napra az aznap órát tartó tanárok számát;

B. azt a tantárgyat, amit a legtöbb tanár tanít;
C. azt a tanárt, akinek a legtöbb lyukasórája van (lyukasóra: aznap előtte is van órája valamikor és utána is van órája valamikor);
D. az adott T tanárt egész héten helyettesíteni tudó tanárt!

A bemenet első sorában a tanárok száma (`1≤N≤100`), a tantárgyak száma (`1≤M≤100`) és egy tanár sorszáma van (`1≤T≤N`). A következő sorok mindegyikében egy tanár sorszáma, tanított tantárgy sorszáma, nap (1 és 5 közötti egész szám), óra (0 és 8 közötti egész szám) található.

Például `3 7 2 0` azt jelenti, hogy a harmadik tanár a hetedik tantárgyat a hét második napján a nulladik órában tanítja.

A kimenetre négy sort kell írni! Az első sorba az A, a másodikba a B, a harmadikba a C, a negyedikbe pedig a D részfeladat eredményét! Ha több megoldás van, akkor az elsőt kell kiírni! Ha nincs megoldás (C és D részfeladatban), akkor -1-et kell kiírni! Az első sorban 5 szám szerepeljen!

Példa:

```
Bemenet:    Kimenet:
3 4 1       2 3 1 0 0
1 1 1 6     2
1 1 2 2     1
1 2 1 3     3
2 1 2 2
2 2 3 1
3 4 1 2
3 2 1 4
3 3 2 1
```

1. tanár	1. nap	2. nap	3. nap	2. tanár	1. nap	2. nap	3. nap	3. tanár	1. nap	2. nap	3. nap
0. óra				0. óra			T2	0. óra			
1. óra				1. óra				1. óra		T3	
2. óra		T1		2. óra		T1		2. óra	T4		
3. óra	T2			3. óra				3. óra			
4. óra				4. óra				4. óra	T2		
5. óra				5. óra				5. óra			
6. óra	T1			6. óra				6. óra			

2. feladat: Újság (30 pont)

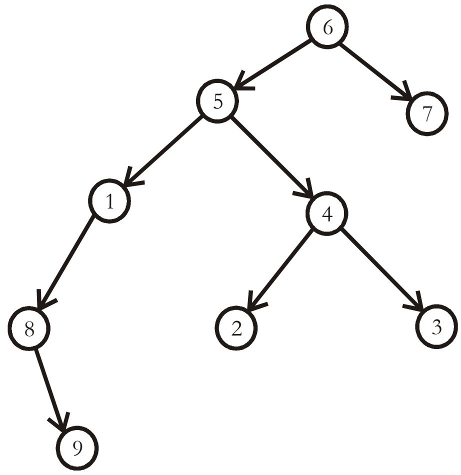
Egy folyóirat terjesztő cég vasúton szállítja minden nap az újságokat a megfelelő címekre. Az újságot egy központi helyen nyomtatják, vonatra rakják és elküldik. A vasúti csomópontokban átrakják a megfelelő irányokba továbbinduló szerelvényekre. Ismerjük minden vasúti csomópontra, hogy közvetlenül honnan kapja az újságcsomagot.

Írj programot, amely megadja:

A. Az adott A csomópontból hány helyre visznek még tovább újságot?
B. Adott A csomópontba küldendő újságokat hányszor kell átrakni másik vonatra?
C. Adott A és B csomópontba küldendő újságokat legtovább melyik csomópontig vihetik együtt?

A bemenet első sorában a csomópontok száma (`1≤N≤1000`), valamint két csomópont sorszáma van (`1≤A,B≤N`). A következő N-1 sor mindegyikében két csomópont sorszáma van (`1≤I≠J≤N`), ami azt jelenti, hogy az I-edik csomópontba a J-edik csomópontból szállítják az újságokat.

A kimenet első sorába az A, a másodikba a B, a harmadikba pedig a C részfeladat eredményét kell írni!

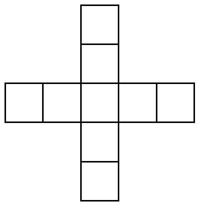


<u>Példa</u>:

Bemenet:	Kimenet:	
`9 1 3`	`2`	{a 8.-ba és a 9.-be}
`1 5`	`1`	{az 5.-ben}
`2 4`	`5`	{az 5.-ig}
`3 4`		
`5 6`		
`7 6`		
`4 5`		
`9 8`		
`8 1`		

<u>3. feladat</u>: Autó (35 pont)

Két út középen keresztezi egymást. Az egyiken csak balról jobbra, a másikon csak felülről lefelé haladhatnak autók. Az útszakaszokat négyzetes cellákra osztjuk, a kereszteződés előtt és mögött is N cella van. A mozgás szabályai:



A. egy autó egy időegység alatt egy cellával mozdulhat el;

Pl.:

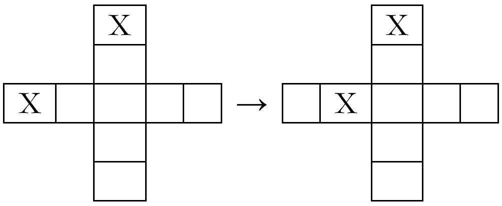
X		X			X		X	→		X		X			X	

B. két autó között mindig kell lenni legalább 1 üres cellának;

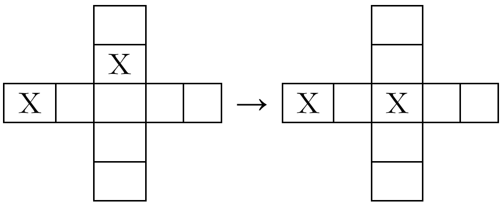
Pl.:

X		X			X		X	

C. a kereszteződésben jobbkéz-szabály van: ha a kereszteződés előtti cellákba egyszerre lépne 2 autó, akkor a balról jövő léphet, a felülről jövő nem;



D. ha a felülről jövő a kereszteződésbe lép, a balról jövőnek tartania kell az 1 cella távolságot.



Készíts programot, amely megadja, hogy az egyes autók mikor jutnak ki az útszakasz végén! (Ha pl. csak balról jobbra jönnek autók, akkor a kilépési idejük pontosan 2*N+1-gyel több, mint a belépési idejük.)

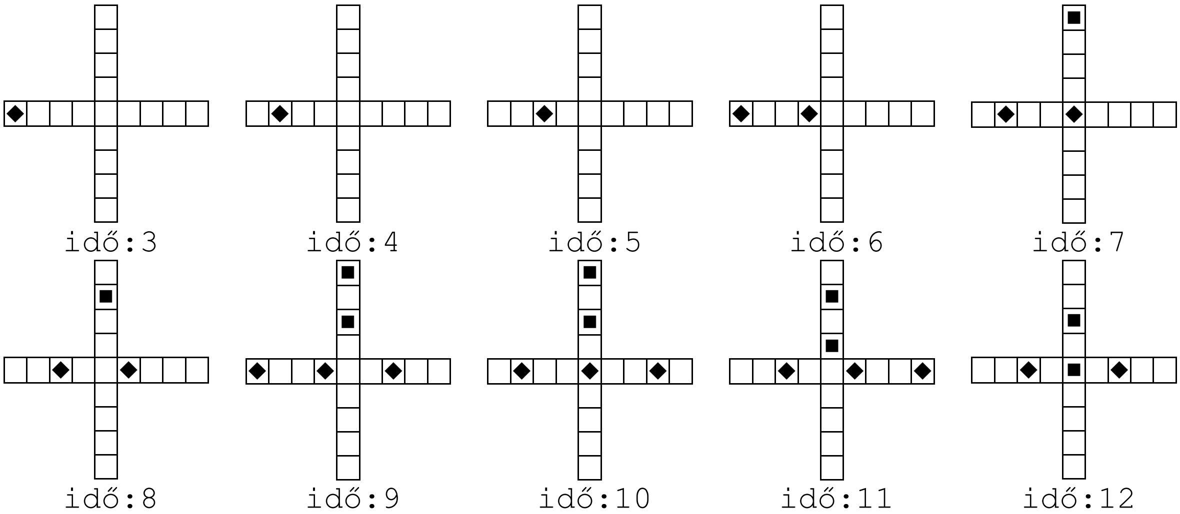
A bemenet első sorában az egyes útszakaszok hossza (`2≤N≤100`), valamint a bejövő autók száma balról és felülről (`0≤B,F≤100`) van. A második sorban `B`, a harmadik sorban `F` belépési idő van (1 és 1000 közötti egész szám), rendre az autók belépési idői balról, valamint felülről.

A kimenet első sorába a jobbra kilépő autók kilépési idejét; a második sorába az alul kilépő autók kilépési idejét kell írni!

Példa:

Bemenet:
```
4 3 2
3 6 9
7 9
```

Kimenet:
```
12 15 19
17 21
```



4. feladat: Járdakövezés (45 pont)

Egy 2 egység széles és N egység hosszú járdát kell kikövezni. A kövezésre 1×2 és 1×3 egység méretű lapokat lehet használni.

Készíts programot, amely kiszámítja, hogy hányféleképpen lehet kikövezni a járdát!

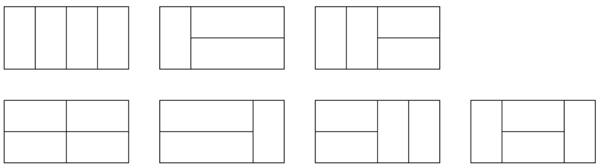
A bemenet első sorában a járda hosszmérete van (1≤N≤50).

A kimenet egyetlen sora azt a számot tartalmazza, ahányféleképpen ki lehet kövezni a járdát!

Példa:

Bemenet:
```
4
```

Kimenet:
```
7
```



Tizenegyedik-tizenharmadik osztályosok

1. feladat: Tanár (32 pont)

Egy iskola tanárairól tudjuk, hogy mikor milyen órát tartanak. A tanárokat, a tantárgyakat, a hét napjait, a napokon belüli órákat sorszámukkal azonosítjuk. Készíts programot, amely megadja:

A. minden napra a szabadnapos tanárok számát;

B. azt a tanárt, akinek a legkevesebb lyukasórája van (lyukasóra: aznap előtte is van órája valamikor és utána is van órája valamikor);

C. az adott T tanárt egész héten helyettesíteni tudó tanárt (ha lehetséges, akkor úgy, hogy szabad napján senkit ne kelljen behívni az iskolába).

D. adott T tanárt a hét H. napján helyettesítő tanárokat úgy, hogy minden óráján szakos helyettesítés legyen!

A bemenet első sorában a tanárok száma (1≤N≤100), a tantárgyak száma (1≤M≤100), egy tanár sorszáma (1≤T≤N) és egy nap sorszáma van (1≤H≤5). A következő sorok mindegyikében egy tanár sorszáma, a tanított tantárgy sorszáma, nap (1 és 5 közötti egész szám), óra (0 és 8 közötti egész szám) van. Például 3 7 2 0 azt jelenti, hogy a harmadik tanár a hetedik tantárgyat a hét második napján, a nulladik órában tanítja.

A kimenetre négy sort kell írni! Az első sorba az A, a másodikba a B, a harmadikba a C, a negyedikbe pedig a D részfeladat eredményé!. Ha több megoldás van, a legkisebb sorszámút kell megadni! Ha nincs megoldás (C és D részfeladatban), akkor -1-et kell kiírni! Az első sorban 5 szám szerepeljen! A negyedik sor első száma a helyettesített órák száma legyen, amit a helyettesítő tanárok sorszámai követnek, órák szerinti sorrendben!

Példa:

```
Bemenet:    Kimenet:

3 4 1 1     1 0 2 3 3
1 1 1 6     2
1 1 2 2     3
1 2 1 3     2 2 2
2 1 2 3
2 2 3 1
3 4 1 2
3 2 1 4
3 3 2 1
```

A példában szereplő 3 tanár órarendje:

1. tanár	1. nap	2. nap	3. nap	2. tanár	1. nap	2. nap	3. nap	3. tanár	1. nap	2. nap	3. nap
0. óra				0. óra			T2	0. óra			
1. óra				1. óra				1. óra		T3	
2. óra		T1		2. óra				2. óra	T4		
3. óra	T2			3. óra		T1		3. óra			
4. óra				4. óra				4. óra	T2		
5. óra				5. óra				5. óra			
6. óra	T1			6. óra				6. óra			

2. feladat: Modul (28 pont)

Egy programrendszer N darab, önállóan lefordítható modult tartalmaz. Minden modulról tudjuk, hogy ki a szerzője, valamint, hogy mely más modulokat használ közvetlenül (ezekre mindenképpen szükség van a lefordításukhoz). A modulokat és a szerzőket is sorszámukkal azonosítjuk.

Írj programot, amely megadja:

A. egy M modul lefordításához mely további modulok lefordítására van szükség (nem csak a közvetlenül használtak kellenek),

B. egy M modul valamely eljárása paraméterei megváltozása miatt mely szerzőket kell felszólítani valamely saját moduljuk megvizsgálására, hogy a modult továbbra is jól használják,

C. egy olyan szerzőt, aki nem használ mások által írt modult.

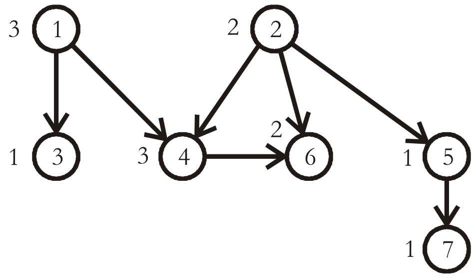
A bemenet első sorában a modulok száma (`1≤N≤1000`), a szerzők száma (`1≤S≤100`), az `M` modulsorszám és a feltételek `F` száma van. A második sorban az i-edik szám az i-edik modul szerzőjének sorszáma. A további `F` sor mindegyikében két modulsorszám szerepel (`1≤I≠J≤N`), ami azt jelenti, hogy az `I`-edik modul fordításához szükség van a `J`-edik modulra.

A kimenet első sorába az A, a másodikba a B, a harmadikba pedig a C részfeladat eredményét kell írni! Az első sor első száma a lefordítandó modulok száma, a többi szám pedig a lefordítandó modulok sorszámai legyenek! A második sor első száma az értesítendő szerzők száma, a továbbiak pedig az értesítendő szerzők sorszámai legyenek! A harmadik sorba egy szerző sorszámot kell írni, ha van megoldás, -1-et, ha nincs megoldás! Több megoldás esetén bármelyik kiírható.

<u>Példa</u>:

Bemenet:	Kimenet:

`7 3 4 7`
`3 2 1 3 1 2 1`
`1 3`
`1 4`
`2 4`
`2 5`
`2 6`
`4 6`
`5 7`
	`1 6`
	`1 2`
	`1`



<u>3. feladat</u>: Zebra (30 pont)

4.		↓		↓	
3.	↓				
2.				↑	
1.		↑		↑	

Egy útszakaszra zebrát festettek az ábrának megfelelően. A zebrát N sorból és M oszlopból álló négyzetráccsal fedtük le. A négyzetrács egyes pontjaiban egyszerre egy ember tartózkodhat. Lentről és fentről is K időegységben érkeznek gyalogosok a szélső cellákba. Egyszerre 1 cella távolságra léphetnek, azaz legalább N+1 időegység kell az úttest túloldalára jutáshoz. A lépésekhez az alábbi szabályokat kell betartani, **a leírás sorrendjében**:

- ha egy gyalogos előtt legalább 2 szabad hely van, akkor előre léphet egyet (akkor is léphet, ha 1 szabad hely és egy vele egy irányban haladó van előtte);

4.										↑	
3.	↓			↑							
2.						⇒	↓	↑		↑	
1.		↑		↑							

- ha valaki előtt vele egy irányba haladó van, és az ellép onnan, akkor a helyére szabad lépni;

4.										↑	
3.				↑						↑	
2.				↑		⇒					
1.											

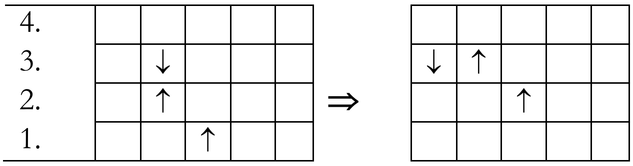
- ha alulról és felülről is ugyanarra a cellára lépne gyalogos, akkor az alulról jövőnek van elsőbbsége, ő léphet, míg a felülről jövő helyben marad;

4.											
3.		↓						↓			
2.						⇒		↑			
1.		↑									

- ha két gyalogos szemben áll egymással, akkor az alulról jövő jobbra kitérhet, azaz jobbra léphet egyet, ha arra a cellára abban az időegységben nem lépne más alulról vagy felülről; ekkor a felülről jövő egyet előre léphet;

4.											
3.		↓									
2.		↑				⇒		↓	↑		
1.											

- ha a két szemben álló közül az alulról jövő nem tud lépni, akkor a felülről jövő jobbra kitérhet (nála a jobbra kitérés persze az ábrán a baloldali szomszédra lépést jelenti), ha arra a cellára abban az időegységben nem lépne más; ekkor az alulról jövő egyet előre léphet;



- ha valaki előtt vele egy irányban haladó áll és nem lép el előle, akkor ő sem léphet.

Készíts programot, amely kiszámítja, hogy az N+1., N+2., …, N+2*K. időegységben hány gyalogos ér át az úttest másik oldalára!

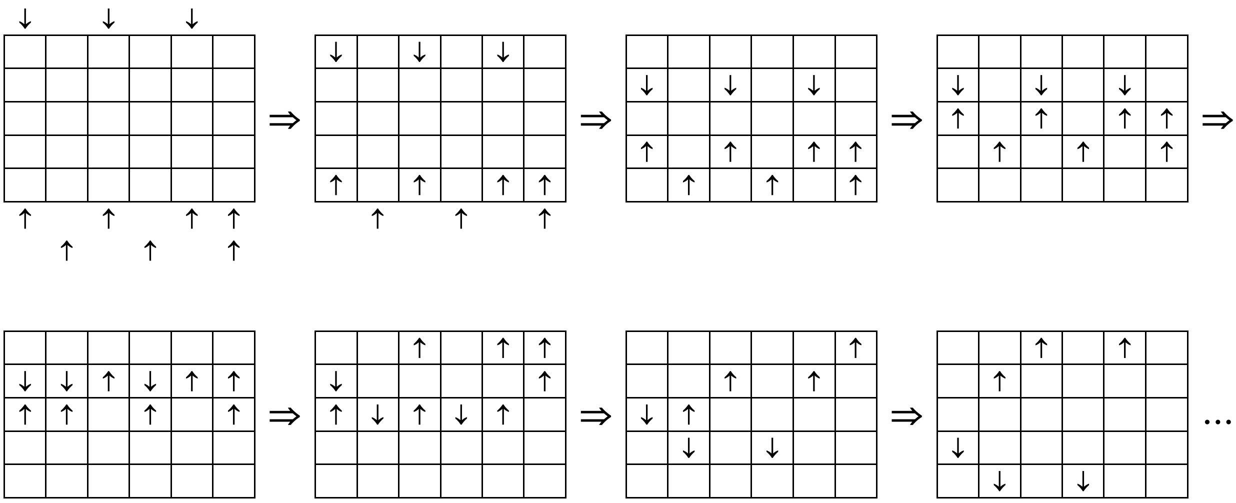
A bemenet első sorában a sorok és oszlopok száma (`1≤N,M≤100`), valamint a vizsgált időegységek száma (`1≤K≤N/2`) van. A következő K sor mindegyike pontosan M bináris számjegyet (0 vagy 1 karakter) tartalmaz. Közülük az i-edik sor j-edik számjegye akkor 1-es, ha az i-edik időegységben az alsó sor j-edik cellájába alulról lép be gyalogos. A következő K sor ugyanilyen adatokat tartalmaz a felülről érkezőkre.

A kimenet 2*K sorába az N+1., N+2., …, N+2*K. időegységben az úttest másik oldalára átérő gyalogosok számát kell kiírni!

Példa:

Bemenet:	Kimenet:
5 6 2	3
101011	1
010101	4
101010	2
000000	

Az ábrán a 0..7 időegységben mutatja a zebrát és a járdát. A zebra a négyzetrács, a járdán várakozók a négyzetrácson kívül láthatók.



4. feladat: Csoportkép (25 pont)

Egy nagyszabású rendezvényre sok vendéget hívtak. Minden vendég előre megadta, hogy mettől meddig lesz jelen a rendezvényen. A szervezők csoportlépeken akarják megörökíteni a résztvevőket. Megbíztak egy fényképészt, hogy készítsen képeket. A fényképész azt vállalta, hogy egy-egy alkalommal T ideig marad, ha az F időpontban érkezik, akkor az F, F+1, …, F+T-1 időpontokban készíthet felvételt, de legfeljebb D alkalommal. Ha a P időpontban készíti a felvételt, akkor azon rajta lesz mindenki, aki akkor jelen van, tehát akinek az E érkezési és U távozási idejére teljesül, hogy E≤P és P<U. A szervezők azt akarják megtudni, hogy legkevesebb hány alkalommal kell kihívni a fényképészt, hogy mindenki rajta legyen legalább egy képen.

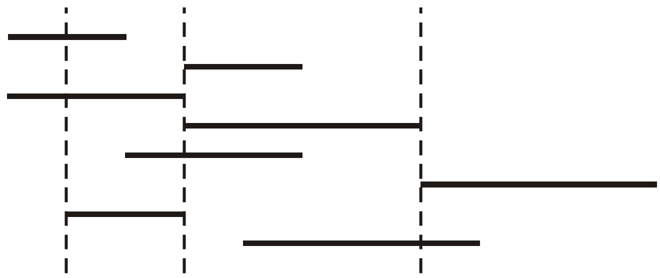
Készíts programot, amely kiszámítja, hogy legkevesebb hány alkalommal kell kihívni a fényképészt, és azt is megadja, hogy mikor!

A bement első sora a vendégek számát (1≤N≤200 000), a fényképész egy alkalommal vállalt tartózkodási idejét (1≤T≤10 000) és az egy alkalommal készíthető képek számát (1≤D≤T) tartalmazza. A további N sor mindegyikében egy-egy vendég érkezési és távozási ideje található (1≤E<U≤20 000).

A kimenetre M+1 sort kell írni! Az első sorban az a legkisebb M szám legyen, ahányszor ki kell hívni a fényképészt, hogy mindenki rajta legyen legalább egy fényképen! A további M sor mindegyike azokat a fényképezési időpontokat tartalmazza növekvő sorrendben, amikor az adott alkalommal fényképet kell készíteni! Több megoldás esetén bármelyik megadható.

Példa:

```
Bemenet:      Kimenet:
8 3 2         2
1 3           2 4
4 6           8
1 4
4 8
3 6
8 12
2 4
5 9
```



5. feladat: Számösszegzés (35 pont)

Adott a_1, …, a_N N pozitív egész szám. Kiszámítandó, hogy a b_1, …, b_M egész számok közül melyek állíthatók elő az a_1, … a_N számok közül vett számok összegeként (egy szám akárhányszor szerepelhet az összegben).

Írj programot, amely megoldja a feladatot!

A bemenet első sora az összegként való előállításban szerepeltethető számok számát (1≤N≤100) és az előállítandó számok számát (1≤M≤10 000) tartalmazza. A második sorban az előállításban szerepeltethető számok vannak ($1 \le S_i \le 30\,000$). A harmadik sor az előállítandó számokat tartalmazza ($1 \le E_i \le 2\,000\,000\,000$).

A kimenet egyetlen sora pontosan M egész számot tartalmazzon! Az i-edik szám értéke 1 legyen, ha a bemenet harmadik sorában az i-edik szám előállítható a bemenet második sorában megadott számok összegeként, egyébként pedig a 0 szám legyen!

Példa:

```
Bemenet:                                       Kimenet:
4 12                                           0 0 1 0 1 1 0 1 1 1 1 1
3 7 13 32
2 5 7 8 9 10 11 12 13 14 15 1234567890
```

2013. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Ember (64 pont)

Ismerjük N emberről, hogy ki szülője kinek (1≤N≤100).

Készíts programot, amely megadja, hogy

A. kinek van a legtöbb gyereke;

B. kinek van a legtöbb unokája;

C. hány embernek nincs unokája!

Példa:

Bemenet:		Kimenet:	Magyarázat:
N=12			
szülő:	gyerek:	A: 4	5,6,7,10 sorszámú
1	2	B: 1	4,8,9,11,12 sorszámú
1	3	C: 10	csak 1-nek és 2-nek van
2	4		
2	8		
2	9		
4	5		
4	6		
4	7		
4	10		
3	11		
3	12		

2. feladat: Tördelés (56 pont)

Egy több bekezdésből álló szöveget úgy kell sorokra tördelni, hogy a bekezdések kezdőbetűjét egy nagy iniciáléval kell kezdeni, ami K sorban L-L karakterhelyet foglal (1≤K, L≤10)! A sorok maximális hossza M (2*L≤M≤80), szavakat nem szabad 2 sorra törni! A szövegben a {} zárójelek között levő részt nem kell kiírni! Ha valamelyik zárójel mégis kellene, akkor abból egymás után 2 szerepel a szövegben – ilyenkor egyet kell kiírni közülük!

Írd programot, amely egyetlen bekezdést olvas be, majd úgy írja sorokra tördelve a szöveget, hogy a bekezdés elején szóközzel tölti fel ezt a K*L helyet, és a kezdőbetűt kihagyja!

Példa:

Bemenet:

M=32, K=4, L=3

```
Egy N kilométer hosszú utat {1<N<10000} rossz állapota miatt
újra aszfaltoznak. Az aszfaltozást M {{M<100}} alkalommal vég-
zik. Minden alkalomról tudjuk, hogy melyik kilométertől melyik
kilométerig terjedő szakaszt aszfaltoznak.
```

Kimenet:

```
    gy N kilométer hosszú utat
    rossz állapota miatt újra
    aszfaltoznak. Az aszfaltozást
    M {M<100} alkalommal végzik.
Minden alkalomról tudjuk, hogy
melyik kilométertől melyik
kilométerig terjedő szakaszt
aszfaltoznak.
```

3. feladat: Út (30 pont)

Egy N kilométer hosszú utat ($1 \le N \le 10000$) rossz állapota miatt szakaszonként újra aszfaltoznak. Az aszfaltozást M alkalommal ($1 \le M \le 100$) végzik. Minden alkalomról tudjuk, hogy melyik kilométertől (K_i) melyik kilométerig (V_i) terjedő szakaszt aszfaltoznak ($0 \le K_i < V_i \le N$).

Készíts programot, amely megadja, hogy hány kilométeren kellene még aszfaltozni, hogy az út teljesen fel legyen újítva!

Példa:

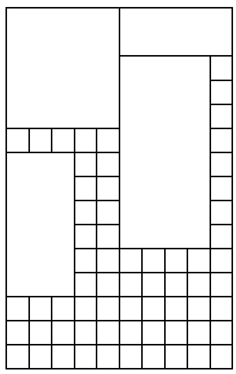
```
Bemenet:
N=100, M=3, aszfaltozás: (10,20), (40,60), (30,50)
Kimenet:
Kell még: 60
Magyarázat: A 0-10, 20-30,  illetve 60-100 kilométer közötti
szakaszokat kell még aszfaltozni!
```

Kilencedik-tizedik osztályosok

1. feladat: Épület (20 pont)

Egy ház tervrajzát egységnyi négyzetrácsos lapon készítik. Minden szobának téglalap alakúnak kell lenni. Eddig N szobát rajzoltak fel a tervrajzon. Minden szobát a bal felső és jobb alsó sarkával adnak meg. A négyzetrács egy mezőjét x-és y-koordinátájával adják meg, a bal felső mező koordinátái (0,0). Az x-koordináták a vízszintesen, az y-koordináták függőlegesen nőnek.

Készíts programot, amely a tervben eddig megadott szobák ismeretében megadja, hogy az épület külső falából hány méter nem tartozik a tervben eddig megadott szobákhoz, valamint hogy a ház négy oldalán hány maximális szakasz van, amelyek nem tartoznak tervben megadott szobák oldalához!



A bemenet első sorában a szobák száma ($1 \le N \le 1\,000\,000$), valamint az épület bal felső (FX,FY) és jobb alsó (AX,AY) sarkának koordinátái vannak ($0 \le FX < AX \le 10\,000$, $0 \le FY < AY \le 10\,000$). A következő N sor mindegyikében egy-egy szoba bal felső (BFX_i, BFY_i) és jobb alsó (JAX_i, JAY_i) sarkának koordinátái vannak ($FX \le BFX_i \le JAX_i \le AX$, $FY \le BFY_i \le JAY_i \le AY$).

A kimenetre két sort kell írni! Az első sorba a külső fal szobákhoz nem tartozó részének hosszát kell írni! A másodikba 4 szám kerüljön: a bal oldali, az alsó, jobb oldali és a felső oldalon levő szobákhoz nem tartozó maximális szakaszok száma!

Példa:

Bemenet:	Kimenet:
4 1 1 10 15	27
1 1 5 5	2 1 1 0
6 1 10 2	
6 3 9 10	
1 7 3 12	

2. feladat: Robot (32 pont)

Egy szennyvíz csatorna hálózathoz takarító robotot fejlesztettek. A hálózat csomópontokból és közöttük levő kör keresztmetszetű csatorna szakaszokból áll, amelyek a szennyvíztisztító felé vezetnek. Ismerjük a csövek átmérőjét. A robot olyan csövet tud tisztítani, amelynek átmérője nagyobb a robot méreténél. A robot csak a tisztító felé, az indulási helyétől a tisztítóig halad.

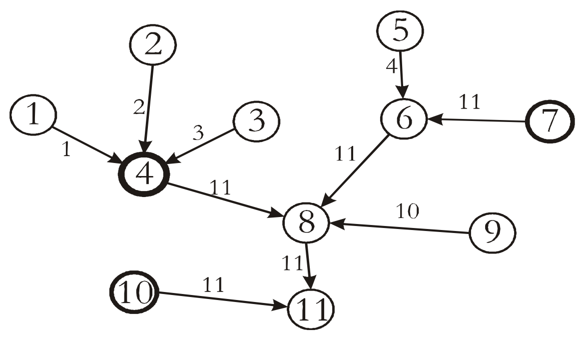
Készíts programot, amely megadja, hogy minimum hány csomópontból kell robotot indítani a tisztító felé, hogy az összes lehetséges csatorna szakaszt kitisztítsák, ahova beférnek, valamint azt, hogy az egyes robotok hány csatorna szakaszt tisztítanak, azaz csatorna szakaszon át érnek a tisztítóig!

A bemenet első sora a pontok számát (1≤N≤1 000 000) és a robot méretét (1≤R≤100) tartalmazza. A következő N-1 sor mindegyike egy-egy csatorna két végpontját ($1 \leq K_i \neq V_i \leq N$) és átmérőjét ($1 \leq A_i \leq 100$) tartalmazza, ahol a szennyvíz a K_i csomópontból a V_i csomópont felé folyik.

A kimenet első sorába a kiindulási csomópontok minimális K számát kell írni! A további K sorba az egyes robotok által kitisztítandó csatorna szakaszok számát kell írni a kiinduló csomópontjuk sorszámának sorrendjében!

Példa:

Bemenet:	Kimenet:
11 10	3
1 4 1	2
2 4 2	3
3 4 2	1
5 6 4	
4 8 11	
6 8 11	
7 6 11	
9 8 10	
8 11 11	
10 11 11	



3. feladat: Hírlánc (32 pont)

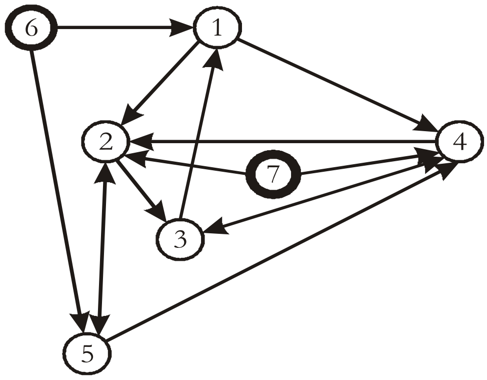
Az osztály tanulói hírláncot alkotnak. Minden tanuló megadja két osztálytársát, akinek mindig továbbítja azt a hírt, amit valakitől kapott. Minden tanuló azonnal továbbadja a kapott hírt, ha azt nem ő indította.

Készíts programot, amely megadja, hogy legkevesebb hány tanulóhoz kell eljuttatni egy új hírt, hogy azt mindenki megkapja!

A bemenet első sorában a tanulók száma van (1≤N≤10 000). A következő N sor mindegyike két egész számot tartalmaz. Közülük az i-edik sor annak a két tanuló sorszámát tartalmazza, akiknek az i-edik tanuló továbbítja a hírt.

A kimenet első sorába azt a legkisebb M számot kell írni, ahány tanulónak el kell juttatni egy új hírt, hogy azt mindenki megkapja! A második sorba kell kiírni annak az M tanulónak a sorszámát, akikhez az új hírt el kell juttatni! Több megoldás esetén bármelyik megadható.

Példa:




```
Bemenet:          Kimenet:
7                 2
2 4               7 6
3 5
4 1
2 3
4 2
1 5
2 4
```

4. feladat: Számok (32 pont)

Adott N pozitív egész szám. Keresünk olyan zárt intervallumokat, hogy minden megadott szám benne legyen valamelyik intervallumban, és minden intervallumba legalább két szám essen és az intervallumok összhossza a lehető legkisebb legyen. Egy [a,b] intervallum hossza a b-a érték.

Készíts programot, amely megadja a legkisebb összhosszú lefedő intervallumokat!

A bemenet első sorában a lefedendő számok száma van ($1<N\leq 100\,000$). A második sor a lefedendő számokat tartalmazza (nem nagyobbak, mint 2 000 000).

A kimenet első sorába a lefedő intervallumok összhosszát kell írni! A következő sorok mindegyikébe egy lefedő intervallum kezdő és végpontját kell írni! Az intervallumokat kezdőpontjuk szerint növekvő sorrendben kell kiírni! Több megoldás esetén bármelyik megadható.

Példa:

```
Bemenet:                 Kimenet:
7                        9
3 1 4 11 7 9 15          1 4
                         7 9
                         11 15
```

5. feladat: Gépek (34 pont)

Egy vállalkozó alkatrészek gyártásával foglalkozik. K különböző fajta alkatrészt tud gyártani két gépén. Mindkét gépe képes legyártani mind a K fajtát, de az egyes fajták legyártása a két gépen különböző ideig tart. Egy időpontban csak az egyik gép dolgozhat a nyersanyag-ellátás miatt. A beérkezett igényeket az érkezés sorrendjében kell kielégítenie. Menet közben átválthat a másik gépre, de az időt igényel.

Készíts programot, amely kiszámítja, hogy a legkevesebb mennyi idő alatt lehet legyártani az összes igényelt alkatrészt!

A bemenet első sorában szerepel az alkatrész fajták száma ($1\leq K\leq 100$), a legyártandó alkatrészek száma ($1\leq N\leq 1000$), az átállás ideje az A gépről a B gépre, továbbá az átállás ideje a B gépről az A gépre. A második sor i-edik száma az i-fajta alkatrész legyártásának ideje az A gépen. A harmadik sor i-edik száma az i-fajta alkatrész legyártásának ideje a B gépen. A negyedik sor tartalmazza a legyártandó alkatrész-fajtákat

A kimenet egyetlen sorába azt a legkisebb időt kell írni, ami alatt a megadott sorrendben legyártható mind az N alkatrész!

Példa:

```
Bemenet:            Kimenet:
3 7 4 3             17
1 3 9
7 1 1
1 2 3 2 1 2 3
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Ház (20 pont)

Egy ház tervrajzát egységnyi négyzetrácsos lapon készítik. Minden szobának téglalap alakúnak kell lenni. Eddig N szobát rajzoltak fel a tervrajzon. Minden szobát a bal felső és jobb alsó sarkával adnak meg. A négyzetrács egy mezőjét x-és y-koordinátájával adják meg, a bal felső mező koordinátái (0,0). Az x-koordináták a vízszintesen, az y-koordináták függőlegesen nőnek. A tervező ki akarja számítani, hogy hány új téglalap alakú szobát lehet még betenni a tervbe, ha bármely két új szoba bármely két oldalának nem lehet közös része, továbbá mind a négy oldala szomszédos vagy meglevő szobával, vagy a ház oldalával. Eddig betervezett szobák olyanok, hogy minden szabadon maradt terület téglalap alakú.

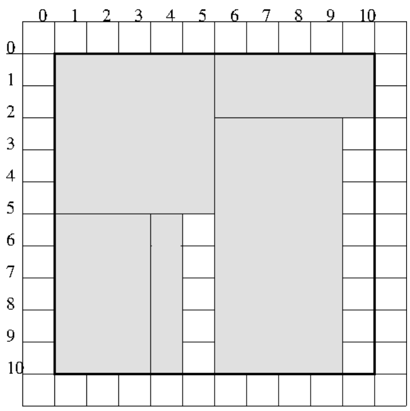
Készíts programot, amely az épület és a tervben meglévő szobák ismeretében megadja, hogy hány téglalap alakú új szobát lehet még betenni a tervbe, valamint mekkora a legnagyobb lehetséges új szoba területe!

A bemenet első sorában a tervben meglevő szobák száma (`1≤N≤1 000 000`), valamint az ház bal felső (`FX,FY`) és jobb alsó (`AX,AY`) sarkának koordinátái vannak (`0≤FX<AX≤10 000`, `0≤FY<AY≤10 000`). A következő N sor mindegyikében egy-egy szoba bal felső (BFX_i,BFY_i) és jobb alsó (JAX_i,JAY_i) sarkának koordinátái vannak ($FX≤BFX_i≤JAX_i≤AX$, $FY≤BFY_i≤JAY_i≤AY$).

A kimenet első sorába a kialakítható új szobák számát kell írni! A második sorba a legnagyobb új szoba területe kerüljön!

Példa:

```
Bemenet:            Kimenet:
5 1 1 10 10         2
1 1 5 5             8
6 1 10 2
6 3 9 10
1 6 3 10
4 6 4 10
```



2. feladat: Csatorna (30 pont)

Egy szennyvíz csatorna hálózathoz takarító robotot fejlesztettek. A hálózat csomópontokból és közöttük levő kör keresztmetszetű csatorna szakaszokból áll, amelyeknek ismerjük a csőátmérőjét. A robot olyan csövet tud tisztítani, amelynek átmérője nagyobb a robot méreténél. A robot a csatornában mindkét irányban haladhat.

Készíts programot, amely megadja, hogy adott pontból indítva a robot hány csatorna szakaszt tud kitisztítani, valamint hogy minimum hány további pontból kellene elindítani, hogy az összes olyan csatornát kitisztítsa, ahova befér!

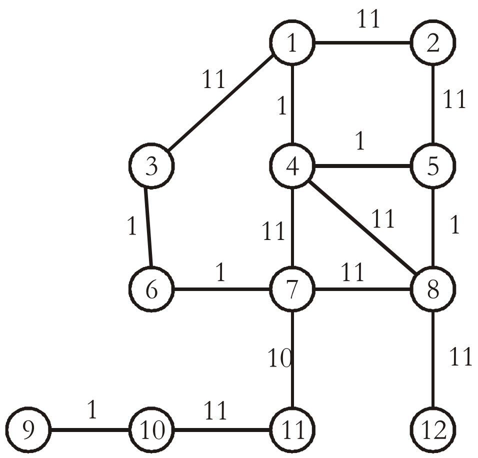
A bemenet első sora a csomópontok számát ($1 \le N \le 1000$), a csatorna szakaszok számát ($1 \le M \le 10\,000$), a kiinduló csomópont sorszámát ($1 \le S \le N$) és a robot méretét ($1 \le R \le 100$) tartalmazza. A következő M sor mindegyike egy-egy csatorna szakasz két végpontját ($1 \le K_i \ne V_i \le N$) és átmérőjét ($1 \le A_i \le 100$) tartalmazza.

A kimenet első sorába az S pontból kitisztítható csatorna szakaszok számát kell írni! A második sorba azon további pontok minimális számát kell írni, amelyekből elindulva az összes olyan csatorna kitisztítható, ahova a robot befér!

Példa:

```
Bemenet:
12 15 4 10
1 2 11
5 8 1
1 3 11
1 4 1
2 5 11
6 7 1
3 6 1
4 7 11
4 8 11
4 5 1
8 7 11
11 7 10
8 12 11
9 10 1
11 10 11
```

```
Kimenet:
4
2
```



3. feladat: Poligon (30 pont)

Adott a síkon egy N csúcspontot tartalmazó zárt, nem metsző törtvonal a csúcsok felsorolásával. Tehát a felsorolásban az i-edik és i+1-edik (i<N) pont van összekötve egyenes szakasszal, illetve az N-edik az elsővel.

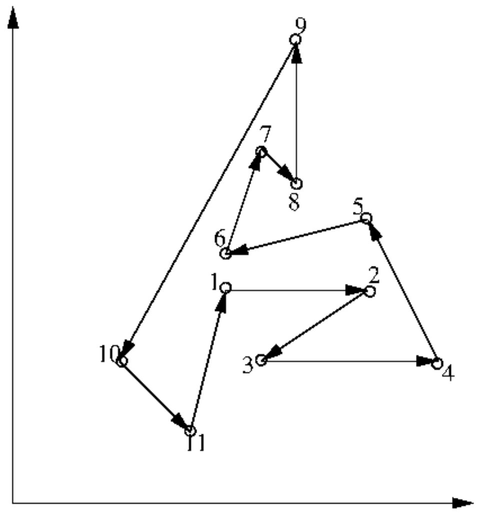
Készíts programot, amely megadja azokat az A,B csúcspárokat, amelyekre teljesül, hogy a törtvonal minden A-tól és B-től különböző pontja szigorúan balra van, ha A-ból B-felé nézünk!

A bemenet első sorában a csúcspontok száma van ($3 \le N \le 10\,000$). A következő N sor mindegyike egy-egy csúcspont x- és y-koordinátáját tartalmazza ($-1\,000\,000 \le X, Y \le 1\,000\,000$).

A kimenet első sorába azon csúcspárok M számát kell írni, amelyekre teljesül a kívánt feltétel! A következő M sor mindegyike egy megfelelő csúcspár A és B sorszámát tartalmazza! A csúcspárokat tetszőleges sorrendben ki lehet írni! A pontpár A és B sorrendje lényeges!

Példa:

Bemenet:	Kimenet:
11	4
6 6	10 11
10 6	11 4
7 4	4 9
12 4	9 10
10 8	
6 7	
7 10	
8 9	
8 13	
3 4	
5 2	



4. feladat: Lefedés (30 pont)

Adott N pozitív egész szám. Keresünk legfeljebb K olyan zárt intervallumot, hogy minden megadott szám benne van valamelyik intervallumban és az intervallumok összhossza a lehető legkisebb. Minden lefedő [a,b] intervallumra teljesülni kell, hogy a<b. Az intervallum hossza a b-a érték.

Készíts programot, amely megadja a legkisebb összhosszú lefedő intervallumokat!

A bemenet első sorában a lefedendő számok száma (1≤N≤100 000) és a lefedésre használható intervallumok számának maximuma (1≤K≤N) van. A második sor a lefedendő számokat tartalmazza (nem nagyobbak, mint 2 000 000).

A kimenet első sorába a lefedő intervallumok összhosszát kell írni! A további legfeljebb K sorba kell kiírni a lefedő intervallumokat, egy sorba egy intervallum kezdő és végpontját! Az intervallumokat kezdőpontjuk szerint növekvő sorrendben kell kiírni! Több megoldás esetén bármelyik megadható.

Példa:

Bemenet:	Kimenet:
7 3	8
3 1 4 11 7 9 15	1 4
	7 11
	15 16

5. feladat: Játék (40 pont)

Ádám és Éva kétszemélyes játékot játszik egy N mezőt tartalmazó játéktáblán. A játék során az első mezőről indulva, felváltva lépve, egy bábut mozgatnak a játéktáblán, ugyanarra a mezőre többször is léphetnek. Egy adott mezőről csak szomszédos mezőre lehet lépni egy lépésben. Minden mezőre rá van írva egy pontszám. Ádám kezdi a játékot. Ha Ádám az aktuális lépésében az m mezőre lép, amire p pontszám van írva, akkor összpontszáma p értékkel növelődik. Ha Éva az m mezőre lép, amire p pontszám van írva, akkor Ádám összpontszámát csökkentik p értékkel. Ádám célja, hogy a lehető legtöbb összpontot szerezze, Éva célja pedig az, hogy Ádám a lehető legkevesebb összpontot szerezzen a játék során. Az összpontszám lehet negatív is, ekkor Ádám fizet Évának a játék végén.

Készíts programot, amely kiszámítja, hogy mekkora az a legnagyobb összpontszám, amit Ádám biztosan meg tud szerezni K lépéses játékban, bárhogyan is lép Éva!

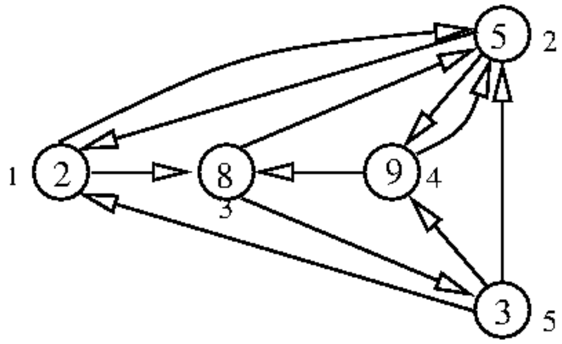
A bemenet első sorában a játéktáblán lévő mezők száma (`1≤N≤500`) és a játékban megteendő lépések száma (`1≤K≤2000`) van. (Tehát mindkét játékos K lépést tesz, felváltva.) A második sor m-edik száma az m sorszámú mezőn lévő pontszám értéke (legfeljebb 1000). A következő N sor írja le a mezők szomszédjait, tehát, hogy adott mezőről mely mezőkre lehet lépni közvetlenül. Közülük az m-edik sor az m-edik mező szomszédjait tartalmazza, 0-val zárva. Minden mezőről legalább egy másik mezőre lehet lépni. Minden mezőnek önmaga is lehet szomszédja.

A kimenet egyetlen sorába a legnagyobb összpontszámot kell írni, amit Ádám meg tud szerezni K lépéses játékban, bárhogyan is lép Éva!

<u>Példa</u>:

```
Bemenet:
5 2
2 5 8 9 3
3 2 0
4 1 0
2 5 0
3 2 0
4 2 1 0
```

```
Kimenet:
4
```



2013. A verseny végeredménye

I. korcsoport

1	Almási Nóra	Fazekas Mihály Gimnázium, Debrecen
2	Baran Zsuzsanna	Fazekas Mihály Gimnázium, Debrecen
3	Alexy Marcell	Juhász Gyula Általános Iskola, Vác
	Pásztor Szabolcs	Fazekas Mihály Gimnázium, Debrecen
5	Bodnár Anna	Szent István Gimnázium, Budapest
6	Radnai László	Veres Péter Gimnázium, Budapest
7	Németh Balázs	Lehel Vezér Gimnázium, Jászberény
8	Szeder Patrik	Széchenyi István Gimnázium, Dunaújváros
9	Tóth Márk Andor	Zrínyi Miklós Gimnázium, Zalaegerszeg
	Augusztin András Balázs	Központi Általános Iskola és Diákotthon, Salgótarján

II. korcsoport

1	Leitereg Miklós	Veres Péter Gimnázium, Budapest
2	Székely Szilveszter	Neumann János Középiskola és Kollégium, Eger
3	Weisz Ambrus	Fazekas Mihály Gimnázium, Budapest
4	Hornák Bence	Berzsenyi Dániel Gimnázium, Budapest
5	Erdős Márton	Batthyány Lajos Gimnázium, Nagykanizsa
6	Végvári Zalán	Fazekas Mihály Gimnázium, Debrecen
7	Ádám Krisztián	Szent László Gimnázium, Budapest
8	Kocsis Dávid	Gábor Dénes Gimn. és Műszaki Szakközépiskola, Szeged
	Almási Péter	Fazekas Mihály Gimnázium, Debrecen
10	Qian Lívia	Ságvári Endre Gimnázium, Szeged

III. korcsoport

1	Weisz Gellért	Fazekas Mihály Gimnázium, Budapest
2	Havasi Márton	Fazekas Mihály Gimnázium, Budapest
3	Szabó Attila	Leőwey Klára Gimnázium, Pécs
4	Somogyvári Kristóf	Ságvári Endre Gimnázium, Szeged
5	Nagy Vendel	Fazekas Mihály Gimnázium, Debrecen
6	Simig Dániel	Fazekas Mihály Gimnázium, Budapest
7	Leitereg András	Veres Péter Gimnázium, Budapest
8	Sipos Szabolcs	Janus Pannonius Gimnázium és SzKI, Pécs
9	Varga Erik Krisztián	Berzsenyi Dániel Gimnázium, Budapest
10	Palkó András	Vörösmarty Mihály Gimnázium, Szentgotthárd

2014. Első forduló

Ötödik-nyolcadik osztályosok

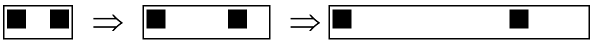
Számítógép nélküli feladatok

<u>1. feladat</u>: Sorminta (40 pont)

Egy sormintában a 0 számjegy jelöli a fehér, az 1 számjegy pedig a fekete elemeket. Például az 1101001 leírásnak a  felel meg.

A sormintát szabályok segítségével állítjuk elő, több lépésen keresztül. Ha van egy sormintát leíró számjegy sorozatunk, akkor arra olyan következő sormintát előállító szabályokat alkalmazhatunk, amelyek megadják, hogy az eredeti sormintában szereplő 0-s és 1-es számjegyek helyére mit (vagy miket) kell írni. Például a kiinduló sorozat az 101, a két szabály: 0 → 00, 1 → 10. Ekkor az 101-ből egy lépésben az 100010 sorozat áll elő, ebből a következő lépésben az 100000001000 lesz:

```
101 ⇒ 100010 ⇒ 100000001000
```



A. Legyen kezdetben a sormintát leíró sorozatban egyetlen számjegy, az 1-es! A két szabály: 0 → 0, 1 → 101. Add meg a szabályok alkalmazásával az első 3 lépésben előállított sormintát (rajzolni is lehet, számjegysorozatot megadni is lehet)!

B. Legyen kezdetben a sormintát leíró sorozatban egyetlen számjegy, az 1-es! A két szabály: 0 → 01, 1 → 10. Add meg a szabályok alkalmazásával az első 4 lépésben előállított sormintát (rajzolni is lehet, számjegysorozatot megadni is lehet)!

C. Legyen kezdetben a sormintát leíró sorozatban egyetlen számjegy, az 1-es! A két szabály: 0 → 11, 1 → 10. Add meg a szabályok alkalmazásával az első 4 lépésben előállított sormintát (rajzolni is lehet, számjegysorozatot megadni is lehet)!

<u>2. feladat</u>: Mit csinál (40 pont)

Az alábbi algoritmus az A, B, C számok ismeretében egyetlen X számot ad eredményül.

```
Valami(A,B,C,X):
  Ha A>B akkor X:=A különben X:=B
  Ha X>C akkor Y:=A+B-X
               Ha Y>C akkor X:=Y különben X:=C
  Elágazás vége
Eljárás vége.
```

A. Mi lesz X értéke, ha A=1, B=2, C=3?

B. Mi lesz X értéke, ha A=20, B=10, C=30?

C. Mi lesz X értéke, ha A=2, B=8, C=4?

D. Mi lesz X értéke, ha A=10, B=100, C=1?

E. Fogalmazd meg általánosan, hogy hogyan függ az eljárás végére érve X értéke a bemenettől!

F. Fogalmazd meg általánosan, hogy milyen értékekre nem teljesül a második elágazás `X>C` feltétele!

<u>3. feladat</u>: Számok (60 pont)

Az alábbi két algoritmus N adatot kap bemenetként az X vektorban (N>2, X(i)>0), amelyekből A és B értékét számolja ki.

Az A, B, C kérdés az első algoritmusra vonatkozik:

```
Valami1(N,X,A,B):
  A:=-1; B:=-2
  Ciklus i=1-től N-ig
    Ha X(i)>A akkor B:=A; A:=X(i)
    különben ha X(i)>B akkor B:=X(i)
  Ciklus vége
Eljárás vége.
```

A. Mi lesz az A és a B változók értéke, ha kezdetben N=5, X=(3,2,5,4,1)?

B. Mi lesz az A és a B változók értéke, ha kezdetben N=5, X=(3,2,5,5,1)?

C. Fogalmazd meg általánosan az egyes kimenő paraméterek szerepét!

A D, E, F kérdés a második algoritmusra vonatkozik:

```
Valami2(N,X,A,B):
  A:=-1; B:=-2
  Ciklus i=1-től N-ig
    Ha X(i)>A akkor B:=A; A:=X(i)
    különben ha X(i)<A és X(i)>B akkor B:=X(i)
  Ciklus vége
Eljárás vége.
```

D. Mi lesz most A és B értéke, ha kezdetben N=5, X=(3,2,5,5,1)?

E. Fogalmazd meg általánosan, miben más most az eredmény, mint az első változatban!

F. Adj olyan helyes bemenetet, amelyre B értéke negatív szám marad!

Számítógépes feladat – VÁLASZTHATÓ

4. feladat: Települések (60 pont)

Ismerjük Magyarország összes településének távolságát Kukutyintól és Piripócstól. Készíts programot, amely megadja

- a Kukutyintól legmesszebb levő település sorszámát (ha több megoldás van, akkor közülük a legkisebb sorszámút);
- azon települések számát, amelyek közelebb vannak Kukutyinhoz, mint Piripócshoz;
- a Piripócshoz 100 kilométernél közelebbi azon települések sorszámát, amelyek Kukutyintól legalább 100 kilométerre vannak (sorszám szerint növekvő sorrendben)!

A bemenet első sorában a települések száma (≥1), alatta soronként egy-egy település távolsága Kukutyintól, illetve Piripócstól. A kimenetre kell kiírni a fenti kérdésekre, feladatokra adott válaszokat!

Minta:

Bemenet (billentyűzet)		Kimenet (képernyő)	
#	Sortartalom	#	Sortartalom [*magyarázat*]
1.	6	1.	4 [*az 1. részfeladat válasza*]
2.	42 15	2.	3 [*a 2. részfeladat válasza*]
3.	110 20	3.	2 6 [*a 3. részfeladat válasza*]
4.	125 160		
5.	166 180		
6.	42 100		
7.	110 39		

Számítógép nélküli feladat – VÁLASZTHATÓ

<u>4. feladat</u>: Települések (60 pont)

Ismerjük Magyarország összes településének távolságát Kukutyintól és Piripócstól. Készítettünk egy algoritmust, amely megadja

A. Piripócs 100 kilométeres körzetében levő települések Piripócstól vett átlagos távolságát;

B. annak a településnek a sorszámát, **amelyen keresztül** a legrövidebb út vezet Kukutyinból Piripócsra (ha több ilyen is van, akkor a legkisebb sorszámút);

C. azon települések Piripócstól vett távolságát, amelyek közelebb vannak Piripócshoz, mint Kukutyinhoz (tetszőleges sorrendben).

Az alábbi eljárásban N a települések száma, a K vektor a Kukutyintól mért távolságokat, a P vektor pedig a Piripócstól mért távolságokat tartalmazza:

```
Település(N,K,P,A,B,C,L)
  A:=0; B:=1; D:=0; L:=0
  Ciklus i=1-től N-ig
    Ha □□□□□□□□□ akkor A:=□□□□□; D:=□□□□□
    Ha □□□□□□□□□ akkor B:=□□□□□
    Ha □□□□□□□□□ akkor L:=□□□□□; C(L):=□□□□□
  Ciklus vége
  Ha □□□□□□□□□ akkor A:=□□□□□ különben A:=0
Eljárás vége.
```

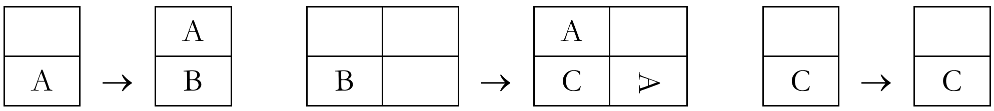
Mit kell írni a téglalapok helyére, hogy a fenti feladatok helyes megoldását kapjuk? (Új sorok nem írhatók, új változó nem vezethető be.) Az A részfeladat megoldása az A változóba, a B-é a B változóba, a C-é az L elemű C tömbbe kerüljön.

Kilencedik-tizedik osztályosok

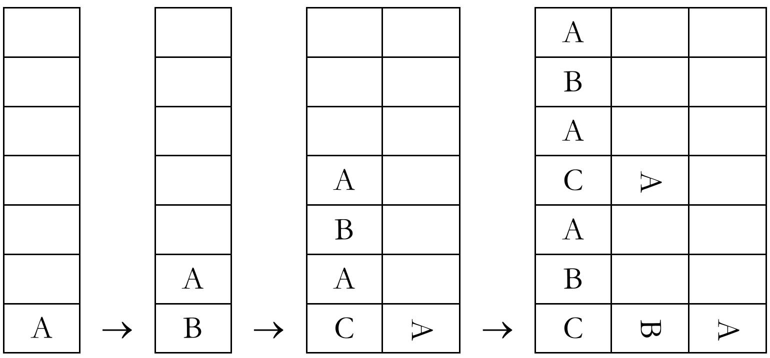
<u>1. feladat</u>: Növény (42 pont)

Egy síkban növekvő növény egyes darabjait az A, B, C betűkkel jelöljük. A növekedés egy időegységében megadjuk minden darabra, hogy a következő időegységben mi lesz belőle. Kezdetben mindig egy darab A típusú elemből áll a növény, amely felfelé néz.

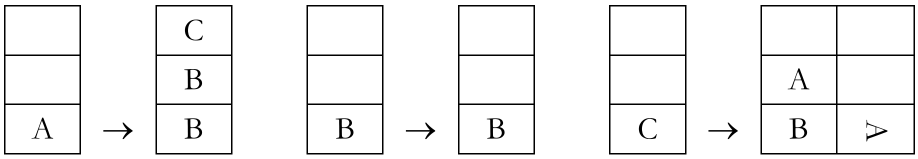
<u>Példa</u>: az A, B és C átalakítási szabályai:



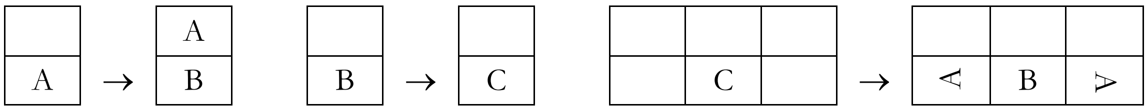
A fenti szabályokkal az egyetlen A típusú elemből álló növény így fejlődik 3 időegységben:



A. Rajzold le az első 4 időegységben a növény fejlődését, ha az alábbi szabályokat kell alkalmazni!



B. Rajzold le az első 4 időegységben a növény fejlődését, ha az alábbi szabályokat kell alkalmazni!



<u>2. feladat</u>: Mit csinál (51 pont)

Az alábbi algoritmus az A, B, C (mindhárom >0) számok alapján számolja ki D értékét.

```
Valami(A,B,C,D):
  Ciklus amíg nem(A=B és B=C)
    Ha A>B akkor A:=A-B
    Ha B>C akkor B:=B-C
    Ha C>A akkor C:=C-A
  Ciklus vége
  D:=A
Eljárás vége.
```

A. Mi lesz D értéke, ha A=3, B=2, C=5? Hány kivonás történik a ciklusban?

B. Mi lesz D értéke, ha A=12, B=4, C=8? Hány kivonás történik a ciklusban?

C. Mi lesz D értéke, ha A=12, B=42, C=18? Hány kivonás történik a ciklusban?

D. Fogalmazd meg általánosan, hogyan függ D értéke a bemenettől!

<u>3. feladat</u>: Adatok (67 pont)

Az alábbi algoritmus N nemnegatív adatot kap bemenetként (`N>2`, `X(1)>0`, `X(N)>0`, `X(i)≥0`) az X vektorban, amelyekből több értéket számol ki:

```
Valami(N,X,A,B,C,E):
  A:=0; B:=0; C:=0; D:=0
  Ciklus i=2-től N-ig
    Ha X(i-1)>0 és X(i)=0 akkor D:=0
    Ha X(i)=0 akkor Ha A=0 akkor A:=i
                  D:=D+1; B:=i
    Elágazás vége
    Ha X(i)>0 és X(i-1)=0 akkor C:=C+1; E(C):=D
  Ciklus vége
Eljárás vége.
```

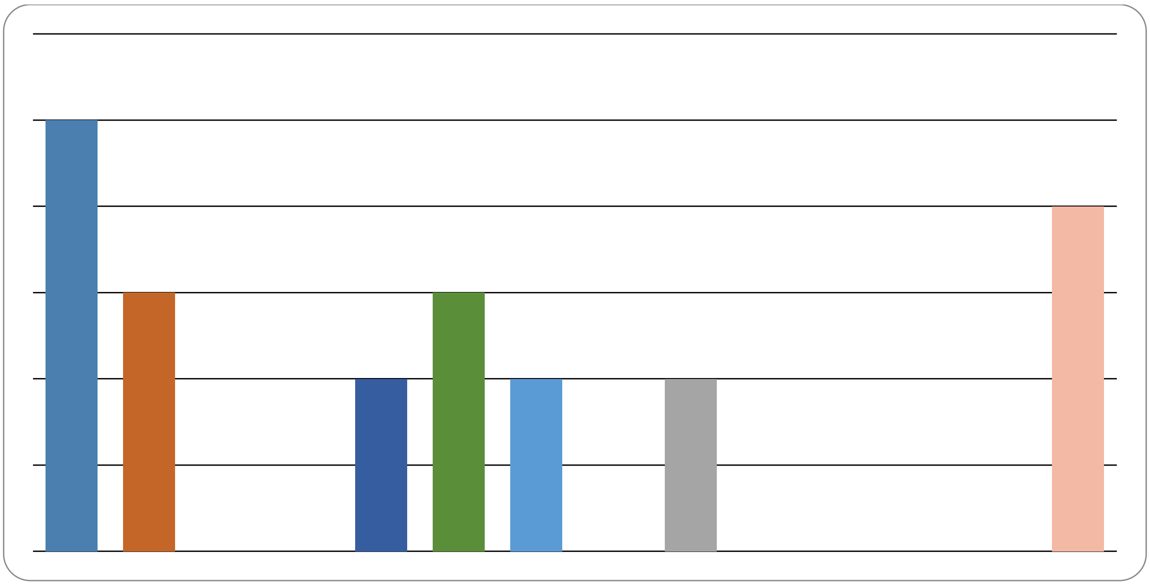
A. Melyik kimenő paraméternek mi lesz az értéke az alábbi 14 elemű X vektor esetén?

B. Fogalmazd meg általánosan az egyes kimenő paraméterek szerepét!

C. Milyen X vektor esetén lehet az eljárás végén A=0 és B=0? (Általános megfogalmazás, vagy konkrét 5 elemű X vektor adható.)

D. Milyen X vektor esetén lehet az eljárás végén A=B>0? (Általános megfogalmazás, vagy konkrét 5 elemű X vektor adható.)

E. Milyen X vektorra lesz adott N esetén C a lehető legnagyobb? (Általános megfogalmazás, vagy konkrét 7 elemű X vektor adható.)



4. feladat: Békák (40 pont)

Egy 2*N+1 hosszú mezőn N barna és N zöld béka áll szemben egymással, közöttük egyetlen üres hellyel. Például N=2 esetén:

B	B		Z	Z

A barna békák jobbra szeretnének eljutni, a zöld békák balra úgy, hogy a mezőt nem hagyják el. Minden lépésben egy béka léphet üres helyre, ami vagy szomszédos, vagy egy béka átugrásával érhető el. A cél az, hogy a békák a lehető legrövidebb idő alatt helyet cseréljenek, azaz a fenti példa esetén így helyezkedjenek el:

Z	Z		B	B

A. Add meg N=1-re, hogy minimum hány lépésben cserélhetnek helyet a békák, és adj meg egy ilyen lépéssorozatot is!

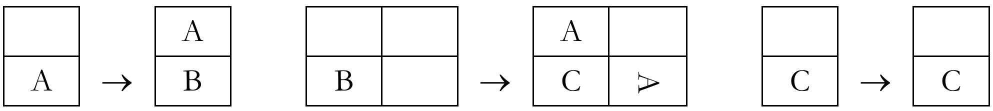
B. Add meg N=2-re, hogy minimum hány lépésben cserélhetnek helyet a békák, és adj meg egy ilyen lépéssorozatot is!

Tizenegyedik-tizenharmadik osztályosok

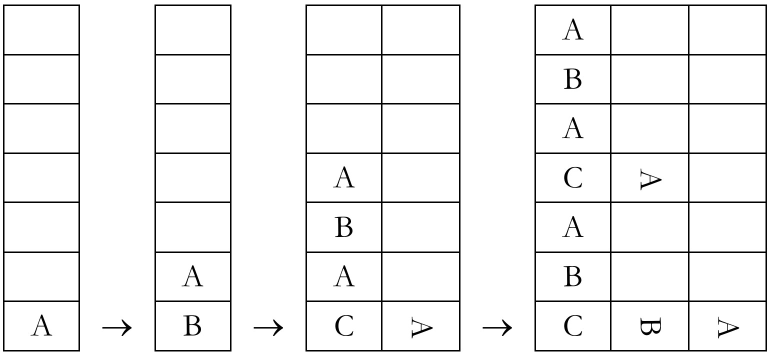
1. feladat: Növény (37 pont)

Egy síkban növekvő növény egyes darabjait az A, B, C betűkkel jelöljük. A növekedés egy időegységében megadjuk minden darabra, hogy a következő időegységben mi lesz belőle. Kezdetben mindig egy darab A típusú elemből áll a növény, amely felfelé néz.

Példa: az A, B és C átalakítási szabályai:



A fenti szabályokkal az egyetlen A típusú elemből álló növény így fejlődik 3 időegységben:

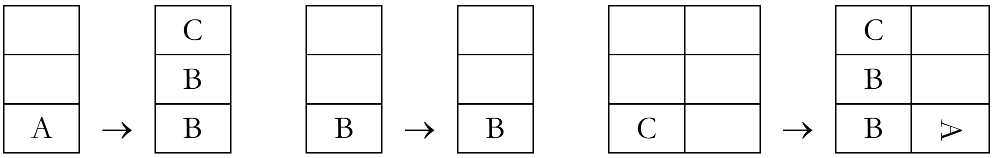


Segítség: A síkbeli rajzokat egyetlen karaktersorozattal is leírhatjuk. Ebben minden betű a növény 1 szakaszát jelöli, az aktuális szakaszból 90 fokkal jobbra ágazó részt () zárójelek, a balra ágazó részt pedig [] zárójelek közé tesszük.

Így a fenti szabályok: A → BA, B → C(A)A, C → C, a zárójeleket pedig a helyükön hagyjuk.

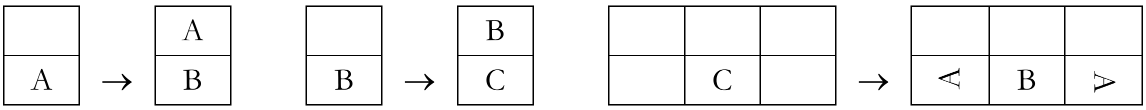
A növény növekedése: A → BA → C(A)ABA → C(BA)BAC(A)ABA

A. Rajzold le az első 4 időegységben a növény fejlődését, ha az alábbi szabályokat kell alkalmazni!



A szabályok karaktersorozattal: A → BBC, B → B, C → B(A)BC

B. Rajzold le az első 4 időegységben a növény fejlődését, ha az alábbi szabályokat kell alkalmazni!



A szabályok karaktersorozattal: A → BA, B → CB, C → B[A](A)

2. feladat: Mit csinál (60 pont)

Az alábbi algoritmus az A, B, C (mindhárom >0) egész számok alapján számolja ki D értékét.

```
Valami(A,B,C,D):
  AA:=A; BB:=B; CC:=C
  Ciklus amíg nem(A=B és B=C)
    Ha A>B akkor B:=B+BB
    Ha B>C akkor C:=C+CC
    Ha C>A akkor A:=A+AA
  Ciklus vége
  D:=A
Eljárás vége.
```

A. Mi lesz D értéke, ha A=3, B=2, C=5? Hány összeadás történik a ciklusban?

B. Mi lesz D értéke, ha A=16, B=4, C=8? Hány összeadás történik a ciklusban?

C. Mi lesz D értéke, ha A=12, B=42, C=18? Hány összeadás történik a ciklusban?

D. Fogalmazd meg általánosan, hogyan függ D értéke a bemenettől!

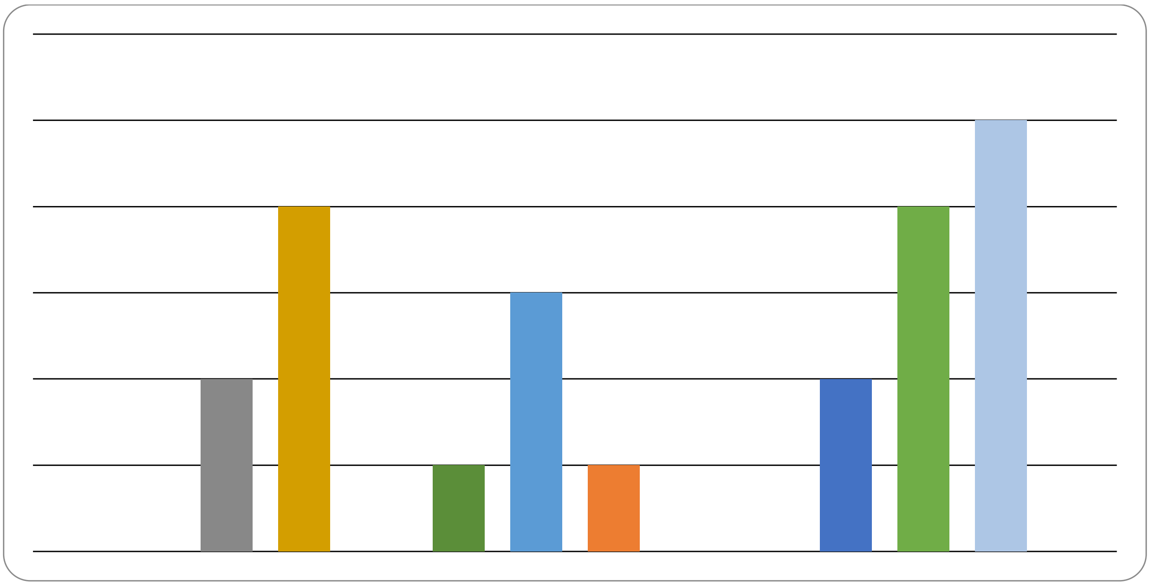
E. Fogalmazd meg képlettel, hogyan függ az összeadások száma A, B, C és a kiszámolt D értékétől!

3. feladat: Adatok (48 pont)

Az alábbi algoritmus N nemnegatív adatot kap bemenetként (N>2, X(1)=0, X(N)=0, X(i)≥0) az X vektorban, amelyekből több értéket számol ki:

```
Valami(N,X,A,B,C):
  A:=0; D:=0; E:=0
  Ciklus i=2-től N-ig
    Ha X(i-1)=0 és X(i)>0 akkor D:=0; E:=0
    Ha X(i)>0 akkor D:=D+1
    Ha X(i)>E akkor E:=X(i)
    Ha X(i)=0 és X(i-1)>0 akkor A:=A+1; B(A):=D; C(A):=E
  Ciklus vége
Eljárás vége.
```

A. Melyik kimenő paraméternek mi lesz az értéke az alábbi 14 elemű X vektor esetén?



B. Fogalmazd meg általánosan az egyes kimenő paraméterek szerepét!

C. Milyen X vektor esetén lehet az eljárás végén A=0? (Általános megfogalmazás, vagy konkrét 5 elemű X vektor adható.)

D. Mi a szerepe a D és az E segédváltozónak?

E. Milyen X vektorra lesz adott N esetén A a lehető legnagyobb? (Általános megfogalmazás, vagy konkrét 7 elemű X vektor adható.)

4. feladat: Békák (25 pont)

Egy N+M+1 hosszú mezőn N barna és M zöld béka áll szemben egymással, közöttük egyetlen üres hellyel.

Például N=2, M=1 esetén:

B	B		Z

A barna békák jobbra szeretnének eljutni, a zöld békák balra úgy, hogy a mezőt nem hagyják el. Minden lépésben egy béka léphet üres helyre, ami vagy szomszédos, vagy egy béka átugrásával érhető el. A cél az, hogy a békák a lehető legrövidebb idő alatt helyet cseréljenek, azaz a fenti példa esetén így helyezkedjenek el:

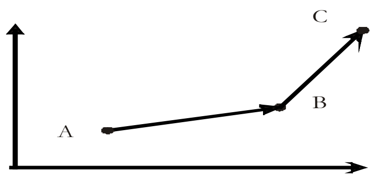
Z		B	B

A. Add meg N=2, M=1-re, hogy minimum hány lépésben cserélhetnek helyet a békák, és adj meg egy ilyen lépéssorozatot is!

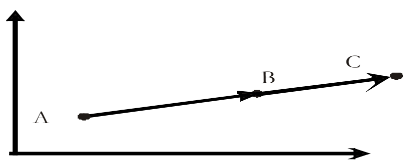
B. Add meg N=2, M=2-re, hogy minimum hány lépésben cserélhetnek helyet a békák, és adj meg egy ilyen lépéssorozatot is!

<u>5. feladat</u>: Geometria (30 pont)

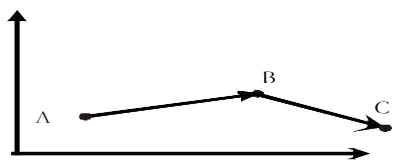
Kaptunk egy függvényt, amelynek három síkbeli pont a paramétere. Csak a következőket tudjuk róla:



`fordul(A,B,C)<0`,
ha A-ból B-n keresztül C-felé haladva B-ben balra kell fordulni.



`fordul(A,B,C)=0`,
ha A-ból B-n keresztül C-felé haladva B-ben egyenesen kell továbbhaladni.

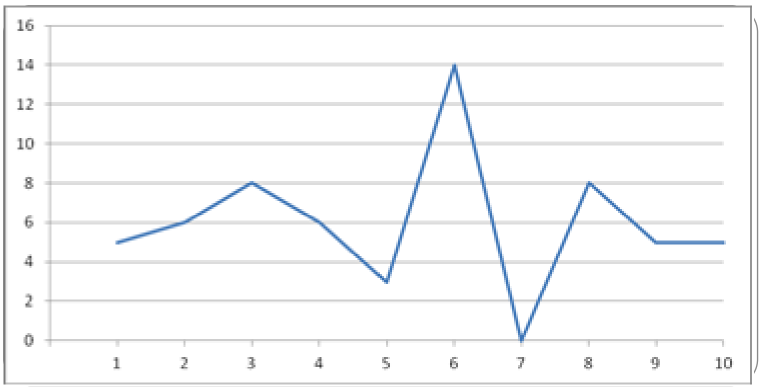


`fordul(A,B,C)>0`,
ha A-ból B-n keresztül C-felé haladva B-ben jobbra kell fordulni.

Mit adnak eredményül a következő algoritmusok?

Add meg Db és Q értékét, ha N=10 és P=((1,5), (2,6), (3,8), (4,6), (5,3), (6,14), (7,0), (8,8), (9,5), (10,5))!

Fogalmazd meg általánosan, mely pontok sorszáma kerül be a Q vektorba a 2. pont után, ha a P vektor a pontokat x-koordináta szerint növekvő sorrendben tartalmazza!




```
Alfa(N,P,Db,Q):
  Db:=1; Q(Db):=2
  Ciklus i=3-tól N-ig
    Ha fordul(P(1),P(i-1),P(i))<0 akkor Db:=Db+1; Q(Db):=i
  Ciklus vége
Eljárás vége.
```

```
Béta(N,P,Db,Q):
  Db:=1; Q(Db):=2; a:=2
  Ciklus i=3-tól N-ig
    Ha fordul(P(1),P(a),P(i))<0
       akkor Db:=Db+1; Q(Db):=i; a:=i
  Ciklus vége
Eljárás vége.
```

```
Gamma(N,P,Db,Q):
  Db:=1; Q(Db):=2; a:=2; b:=2
  Ciklus i=3-tól N-ig
    Ha fordul(P(1),P(a),P(i))<0
       akkor Db:=Db+1; Q(Db):=i; a:=i
    Ha fordul(P(1),P(b),P(i))>0
       akkor Db:=Db+1; Q(Db):=i; b:=i
  Ciklus vége
Eljárás vége.
```

2014. Második forduló

Ötödik-nyolcadik osztályosok

1. feladat: Mértékegységek (40 pont)

Az angol-amerikai mértékegységrendszerben hosszmértéknek a mienktől eltérő rendszert is használnak.

mérföld	≡ 1760 yard
yard	≡ 3 láb
foot (láb)	≡ 1/3 yard
inch (hüvelyk)	≡ 1/12 láb

Készíts programot, amely egy angol mértékegységrendszerben megadott hosszúságot összead, illetve kivon!

A bemenet két sorában a két – pozitív – hosszúság van: a mérföld, a yard, a láb és a hüvelyk értéke egy-egy szóközzel elválasztva.

A kimenetre 2 sort kell írni, az első sorba az összegüket, a másodikba a különbségüket (mindig az elsőből vonjuk ki a másodikat), a bemenet szerinti formátumban!

Példa:

Bemenet (billentyűzet)		Kimenet (képernyő)	
#	Sortartalom [*magyarázat*]	#	Sortartalom [*magyarázat*]
1.	2 850 2 7 [*első hosszúság*]	1.	4 21 0 11 [*összegük*]
2.	1 930 1 4 [*második hosszúság*]	2.	0 1680 1 3 [*különbségük*]

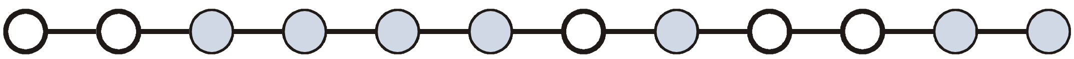
2. feladat: Kínai Nagy Fal (55 pont)

A Kínai Nagy Falon N őrhelyet létesítettek. Közülük azonban nincs mindenhol őrség. Védettnek nevezzük azt a két szomszédos őrhely közötti falat, amelynek mindkét végén van őrség. Őrzöttnek nevezzük azt a két szomszédos őrhely közötti falat, amelynek legalább az egyik végén van őrség.

Készíts programot, amely megadja a védett és az őrzött falak számát, valamint a nem védett falakból álló leghosszabb folytonos sorozat hosszát!

A bemenet első sorában az őrhelyek száma (1≤N≤100) van. A következő N sor az őrségek leírását tartalmazza, közülük az i-edik 0, ha az i-edik őrhelyen nincs őrség; 1 pedig akkor, ha van.

A kimenetre három sort kell írni! Az első sorba a védett falak száma, a másodikba az őrzött falak száma kerüljön! A harmadik sorba a nem védett falakból álló leghosszabb folytonos sorozat hosszát kell írni!

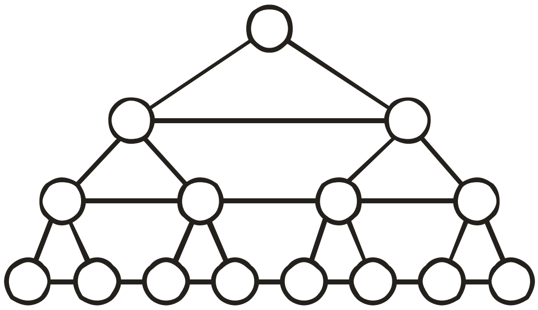


Példa:

Bemenet (billentyűzet)		Kimenet (képernyő)	
#	Sortartalom [*magyarázat*]	#	Sortartalom [*magyarázat*]
1.	12 [*az őrhelyek száma*]	**1.**	4 [*a védett falak száma: 3-4, 4-5, 5-6, 11-12 közötti*]
2.	0	**2.**	9 [*az őrzött falak száma: csak az 1-2 és a 9-10 közötti nem*]
3.	0		5 [*a leghosszabb nem védett sorozat: 6-11 közötti*]
4.	1		
5.	1		
6.	1		
7.	1		
8.	0		
9.	1		
10.	0		
11.	0		
12.	1		
13.	1		

3. feladat: Játék (55 pont)

Egy játéktábla 11 sorból áll, minden sorában pontosan kétszer annyi elem van, mint a fölötte levő sorban. A tábla a következő szerkezetű:



A tábla felső pontjából, azaz az első sor első pontjából indulunk. Az egyes lépéseket a következők írják le:

BL balra lefelé lépünk egyet,

JL jobbra lefelé lépünk egyet,

F felfelé lépünk egyet,

B balra lépünk egyet,

J jobbra lépünk egyet.

Készíts programot, amely beolvas egy lépéssorozatot, majd megadja, hogy a lépéssorozat végén hányadik sor hányadik elemén áll a bábunk!

A bemenet első sorában a lépések száma van (1≤K≤10), a következő K sorban pedig az egyes lépéseket leíró betűk vagy betűpárok.

A kimenet egyetlen sorába a lépéssorozat végén a bábu sorának és soron belül helyének sorszámát kell írni! Ha a bábu kilépne a tábláról, akkor az első sorba azt a helyet kell írni, ahol utoljára volt!

Példa:

	Bemenet (billentyűzet)		Kimenet (képernyő)
#	Sortartalom [*magyarázat*]	#	Sortartalom [*magyarázat*]
1.	6 [*a lépések száma*]	1.	4 7 [*a 4. sor 7. elemére ért a bábu*]
2.	BL		
3.	JL		
4.	J		
5.	F		
6.	JL		
7.	BL		

Kilencedik-tizedik osztályosok

1. feladat: Fal (44 pont)

A Kínai Nagy Falon N őrhelyet létesítettek. Közülük azonban csak M helyen van őrség. Két szomszédos őrhely közötti fal őrzött, ha legalább az egyik végén van őrség; védett, ha mindkét végén van őrség. Őrzött szakasznak nevezzük egymást követő őrzött falak nem bővíthető sorozatát. Hasonlóan, védett szakasznak nevezzük egymást követő védett falak nem bővíthető sorozatát.

Készíts programot, amely megadja a védett és az őrzött szakaszok számát, valamint azt, hogy minimum hány helyre kell még őrséget küldeni, hogy minden fal őrzött legyen!

A bemenet első sorában az őrhelyek száma (1≤N≤100) és az őrségek száma (1≤M≤100) van. A következő M sor az őrségek leírását tartalmazza, közülük az i-edik annak az őrhelynek a sorszáma, ahol az i-edik őrség van. Tudjuk, hogy minden helyen legfeljebb 1 őrség van.

A kimenetre három sort kell írni! Az első sorba a védett szakaszok száma, a másodikba az őrzött szakaszok száma kerüljön! A harmadik sorba az új őrségek minimális számát kell írni, amivel elérhető, hogy minden fal őrzött legyen!

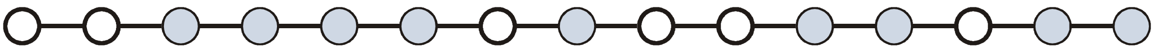
Példa:

Bemenet:

```
15 9
6
3
12
11
4
5
8
15
14
```

Kimenet:

```
3 [3-6. őrhely, 11-12. őrhely, 14-15. őrhely]
2 [2-9. őrhely, 10-15. őrhely]
2 [az 1. és a 9. őrhelyre]
```



2. feladat: Utcák (30 pont)

Egy modern nagyváros úthálózata egy négyzetráccsal írható le, ahol N jelöli a négyzetrács sorainak számát (azaz a kelet-nyugati irányú utak számát, az ilyen utakat alulról felfelé sorszámozzuk), M pedig az oszlopokét (azaz az észak-déli utakét, az ilyen utakat balról jobbra sorszámozzuk). El szeretnénk jutni a város egyik kereszteződéséből egy másik kereszteződésbe. Az egyes kereszteződésekből kivezető néhány út elejére behajtani tilos táblákat helyeztünk el, arra értelemszerűen nem lehet haladni.

Útközben a várost nem hagyhatjuk el (bár erről szóló jelzőtáblák nincsenek).

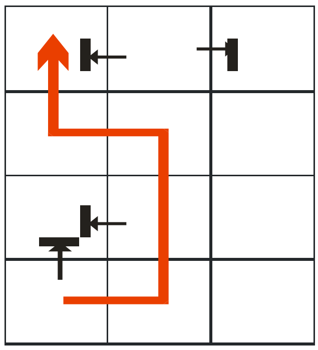
Írj programot, amely a táblák figyelembevételével megadja a legrövidebb útvonalat, amelyeken áthaladva eljuthatunk az indulási helyről a célba!

A bemenet első sorában a sorok és oszlopok száma ($1 \le N, M \le 100$), valamint a táblák száma ($1 \le T \le 10\ 000$) van. A következő T sorban soronként egy-egy tábla leírása található. A táblaleírás formája: sor_1 $oszlop_1$ sor_2 $oszlop_2$, ahol a sor_i és az $oszlop_i$ két szomszédos csomópont koordinátáit adja meg ($1 \le sor_i \le N$, $1 \le oszlop_i \le M$). Jelentése: a (sor_1, $oszlop_1$)-ből a (sor_2, $oszlop_2$)-be vezető útra behajtani tilos tábla van. Az utolsó sorban a két kereszteződés sor és oszlopindexe van, az első az induló hely, a második a cél.

A kimenet első sorába a két pont közötti legrövidebb út hosszát kell írni! A második sorba egy legrövidebb út leírását kell írni, ahol minden lépést a haladás iránya, azaz az E, K, D vagy N betű azonosít! Feltehető, hogy ilyen út mindig van!

Példa:

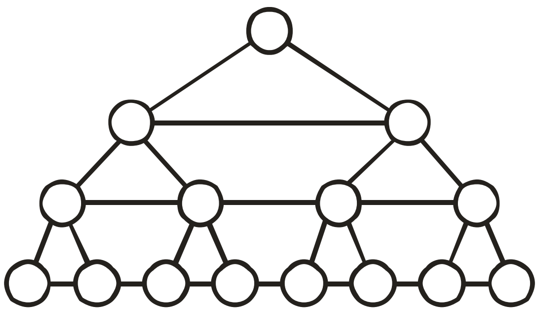
Bemenet:	Kimenet:
4 3 4	5
1 1 2 1	KEENE
2 2 2 1	
4 2 4 1	
4 2 4 3	
1 1 4 1	



1 1 2 1 magyarázata: Az (1,1) pontból a (2,1) pont felé vezető úton behajtani tilos tábla van, arra nem mehetünk.

3. feladat: Játéktábla (40 pont)

Egy játéktábla 101 sorból áll, minden sorában pontosan kétszer annyi elem van, mint a fölötte levő sorban. A tábla a következő szerkezetű:



A tábla felső pontjából indulunk. Az egyes lépéseket a következők írják le:

- 0 balra lefelé lépünk egyet,
- 1 jobbra lefelé lépünk egyet,
- 2 felfelé lépünk egyet,
- 3 balra lépünk egyet,

- 4 jobbra lépünk egyet.

Készíts programot, amely beolvas egy lépéssorozatot, amely elvezet a tábla valamely eleméhez, majd megad egy olyan lépéssorozatot, amely a legrövidebb úton vezet ugyanide!

A bemenet első sorában a lépések száma van (1≤K≤100), a következő sorban pedig az egyes lépéseket leíró K darab szám. A lépéssorozat biztosan helyes, azaz nem hagyjuk el vele a játéktáblát.

A kimenet első sorába a legrövidebb lépéssorozat L hosszát kell írni, amely a bemenetben kapott lépéssorozattal azonos helyre vezet! A második sorba pedig egy ilyen legrövidebb lépéssorozat kerüljön, azaz L szám!

Példa:

Bemenet:	Kimenet:
6	3
0 1 4 2 1 0	1 1 0

4. feladat: Mozi (36 pont)

Egy nagyon várt film vetítésére a szervező jegyrendeléseket fogad. Minden igénylő egy jegyet igényelhet, az igénylésben megad egy ülőhely sorszámot. A feltétel az, hogy ha egy igénylő az igényében az s sorszámot adta meg, akkor el kell fogadnia olyan u sorszámú ülőhelyet, amelyre teljesül, hogy s≤u≤s+K, ahol K egy előre rögzített nemnegatív szám. A szervező feladata, hogy az igénylések közül kiválasztja a legtöbb igényt, amelyet ki tud elégíteni. Bármely ülőhelyet legfeljebb egy igénylő kaphat meg.

Készíts programot, amely kiszámítja, hogy legjobb esetben hány igénylő kérését lehet kielégíteni! A program adjon is meg egy megfelelő jegykiosztást!

A bemenet első sorában az ülőhelyek száma (1≤M≤3000), az igények száma (1≤N≤10 000) és a K értéke (0≤K≤100) van. A második sor i-edik száma annak az ülőhelynek a sorszáma, amelyet az i-edik igénylő szeretne megkapni.

A kimenet első sora egy L egész számot tartalmazzon, a legtöbb kielégíthető igény számát! A következő L sor egy megfelelő jegykiosztást tartalmazzon! Minden sor első száma egy igénylő sorszáma, a második pedig annak az ülőhelynek a sorszáma legyen, amelyiket ez az igénylő kap! A kiírás sorrendje tetszőleges. Több megoldás esetén bármelyik megadható.

Példa:

Bemenet:	Kimenet:
5 7 1	5
4 2 1 3 2 4 5	3 1
	2 2
	5 3
	4 4
	1 5

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Nagy Fal (38 pont)

A Kínai Nagy Falon N őrhelyet létesítettek. Közülük azonban csak M helyen van őrség. Két szomszédos őrhely közötti fal őrzött, ha legalább az egyik végén van őrség; védett, ha mindkét végén van őrség. Őrzött szakasznak nevezzük egymást követő őrzött falak nem bővíthető sorozatát. Hasonlóan, védett szakasznak nevezzük egymást követő védett falak nem bővíthető sorozatát. Ha egy helyen több őrség is van, akkor a feleslegesek elküldhetők más őrhelyre (egynek azonban ott kell maradnia).

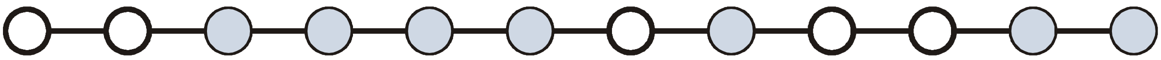
Készíts programot, amely megadja a védett és az őrzött szakaszok számát, valamint azt, hogy a felesleges őrségek jó helyre küldésével hány további fal tehető védetté, illetve őrzötté! (Az utóbbi két feladathoz lehetséges, hogy a felesleges őrségeket más-más helyre kellene küldeni.)

A bemenet első sorában az őrhelyek száma ($1 \le N \le 100$) és az őrségek száma ($1 \le M \le 100$) van. A következő M sor az őrségek leírását tartalmazza, közülük az i-edik annak az őrhelynek a sorszáma, ahol az i-edik őrség van.

A kimenetre négy sort kell írni! Az első sorba a védett szakaszok száma, a másodikba az őrzött szakaszok száma kerüljön! A harmadik sorba a védetté, a negyedik sorba az őrzötté tehető újabb falak számát kell írni!

Példa:

Bemenet:	Kimenet:	
12 9	2	[a 3-6 és a 11-12 között]
6	2	[a 2-9 és 10-12 közötti]
3	3	[pl. a 6-7,7-8 és 2-3 közötti]
12	2	[az 1-2 és a 9-10 közötti]
11		
4		
5		
8		
6		
6		



2. feladat: Város (26 pont)

Egy modern nagyváros úthálózata egy négyzetráccsal írható le, ahol N jelöli a négyzetrács sorainak számát (azaz a kelet-nyugati irányú utak számát, az ilyen utakat alulról felfelé sorszámozzuk), M pedig az oszlopokét (azaz az észak-déli utakét, az ilyen utakat balról jobbra sorszámozzuk). El szeretnénk jutni a város egyik kereszteződéséből egy másik kereszteződésbe. Az egyes kereszteződések (csomópontok) előtt a haladási irányt befolyásoló jelzőtáblák lehetnek, melyeket a következő kódokkal látunk el:

- balra fordulni tilos BT
- jobbra fordulni tilos JT
- balra haladni kötelező BK
- jobbra haladni kötelező JK

Visszafordulni semelyik csomópontban sem lehet, azaz ha nincs jelzőtábla, akkor is csak maximum háromfelé haladhatunk. Útközben a várost nem hagyhatjuk el (bár erről szóló jelzőtáblák nincsenek). Az indulási helyről bármerre indulhatunk, azt nem befolyásolja tábla.

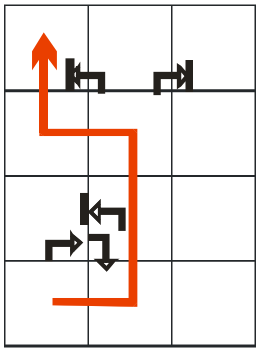
Írj programot, amely a táblák figyelembevételével megadja a legrövidebb útvonalat, amelyeken áthaladva eljuthatunk az indulási helyről a célba!

A bemenet első sorában a sorok és oszlopok száma ($1 \le N,M \le 100$), valamint a táblák száma ($1 \le T \le 10\,000$) van. A következő T sorban soronként egy-egy tábla leírása található. A táblaleírás formája: `irány sor oszlop kód`, ahol a `sor` és az `oszlop` a csomópont koordinátáit adja meg ($1 \le sor \le N$, $1 \le oszlop \le M$), az `irány` pedig a csomópontba beérkező útszakasz irányát (E, K, D, N), azaz a beérkező út északi, keleti, déli vagy nyugati irányban halad-e. Az utolsó sorban a két kereszteződés sor és oszlopindexe van, egy-egy szóközzel elválasztva, az első az induló hely, a második a cél.

A kimenet első sorába a két pont közötti legrövidebb út hosszát kell írni! A második sorba a legrövidebb út leírását kell írni, ahol minden lépést a haladás iránya, azaz az E, K, D vagy N betű azonosít! Feltehető, hogy ilyen út mindig van!

Példa:

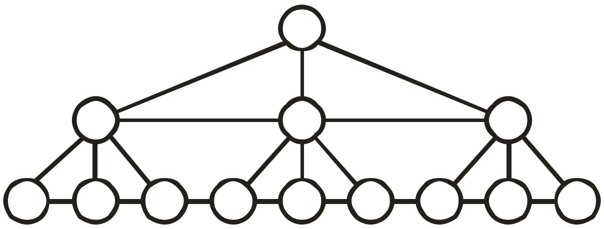
Bemenet:	Kimenet:
`4 3 5`	`5`
`E 2 1 JK`	`KEENE`
`K 2 2 JK`	
`E 2 2 BT`	
`E 4 2 BT`	
`E 4 2 JT`	
`1 1 4 1`	



`E 2 1 JK` magyarázata: Ha északi irányba haladva a (2,1) pontba érünk, akkor ott kötelező jobbra fordulni, azaz innen csak keleti irányba a (2,2) pontba mehetünk.

3. feladat: Játéktábla (26 pont)

Egy játéktábla 101 sorból áll, minden sorában pontosan háromszor annyi elem van, mint a fölötte levő sorban. A tábla a következő szerkezetű:



A tábla felső pontjából indulunk. Az egyes lépéseket a következők írják le:

- 0 balra lefelé lépünk egyet,
- 1 középen lefelé lépünk egyet,
- 2 jobbra lefelé lépünk egyet,
- 3 felfelé lépünk egyet,
- 4 balra lépünk egyet,
- 5 jobbra lépünk egyet.

Készíts programot, amely beolvas egy lépéssorozatot, amely elvezet a tábla valamely eleméhez, majd megad egy olyan lépéssorozatot, amely a legrövidebb úton vezet ugyanide!

A bemenet első sorában a lépések száma van (`1≤K≤100`), a következő sorban pedig az egyes lépéseket leíró K darab szám. *A lépéssorozat biztosan helyes, azaz nem hagyjuk el vele a játéktáblát.*

A kimenet első sorába a legrövidebb lépéssorozat L hosszát kell írni, amely a bemenetben kapott lépéssorozattal azonos helyre vezet! A második sorba pedig egy ilyen legrövidebb lépéssorozat kerüljön, azaz L szám!

Példa:

Bemenet:	Kimenet:
`6`	`2`
`0 2 5 3 1 5`	`1 2`

4. feladat: Gépek (25 pont)

Egy vállalkozó a következő N napra megrendeléseket fogad. Minden munkát egy nap alatt tud elvégezni, amihez egy gépet használ. M megrendelés érkezett, minden megrendelésben szerepel, hogy az igényelt munkát milyen határidőig kell elvégezni. A vállalkozónak ki kell számítani, hogy legkevesebb hány gépre van szükség, hogy minden igényelt munkát határidőre el tudjon végezni.

Készíts programot, amely kiszámítja, hogy legkevesebb hány gép kell ahhoz, hogy minden megrendelt munkát határidőre el tudjon végezni a vállalkozó! A program adja is meg, hogy ez egyes megrendelést melyik napon, melyik gépen végezze el a vállalkozó!

A bemenet első sora a munkanapok N számát ($1 \le N \le 10\,000$), valamint a megrendelések számát ($1 \le M \le 100\,000$) tartalmazza. A második sor i-edik száma az i-edik megrendelés határideje ($1 \le h_i \le N$).

A kimenetre M+1 sort kell írni! Az első sor a minimálisan szükséges gépek G számát tartalmazza! A további M sor tartalmazza az igényelt munkák beosztását az igények sorszáma szerinti sorrendben! Minden sor első száma annak a napnak a sorszáma legyen, amelyik napon a munkát elvégzik, a második szám pedig annak a gépnek a sorszáma legyen, amelyiken a munkát elvégzik! Több megoldás esetén bármelyik megadható.

Példa:

Bemenet:	Kimenet:
10 8	2
3 2 3 2 4 5 6 2	2 2
	1 1
	3 1
	2 1
	3 2
	4 1
	4 2
	1 2

5. feladat: Fazekas (35 pont)

Korondi János fazekas mester két kemencében égeti ki a termékeit. A kiégetésre váró tárgyak az elkészülés sorrendjében várakoznak a kiégetésre. Minden tárgyra rá van írva, hogy legkevesebb hány percig kell égetni a kemencében. Mindkét kemencébe egyidejűleg legfeljebb K darab tárgy rakható kiégetésre. Minden tárgyat az elkészülés sorrendjében betesz valamelyik kemencébe, mindkettőbe legalább egy tárgyat kell rakni egy menetben és a következő menet csak akkor kezdődik, ha mindkét kemence befejezte az égetést. Ha valamelyik kemencében több tárgyat éget egy menetben, akkor az égetési idő az abba a kemencébe tett tárgyak egyedi égetési idejének a maximuma. Mivel a kemencék energia fogyasztása az égetési idő függvénye, ezért a fazekasnak arra kell törekednie, hogy az összesített égetési idő a lehető legkevesebb legyen.

Írj programot, amely kiszámítja, hogy legkevesebb mennyi összesített égetési idő alatt lehet kiégetni az összes tárgyat!

A bemenet első sorában a kiégetésre váró tárgyak száma ($2 \le N \le 1000$) és a két kemence egyedi kapacitása ($2 \le K \le 20$) van, azaz egyidejűleg legfeljebb K tárgy rakható mindkét kemencébe. A második sor i-edik száma az i-edik tárgy minimális égetési ideje (legfeljebb 20 000).

A kimenet első sorába a legkevesebb összesített égetési időt kell írni! A következő N sor mindegyike két egész számot tartalmazzon: annak a menetnek a sorszámát, amelyikben az i-edik tárgyat kemencébe rakjuk, illetve 1, ha az első, 2, ha a második kemencébe rakjuk égetni! A kiírást a menetek sorrendjében kell megadni! Több megoldás esetén bármelyik megadható.

<u>Példa</u>:

Bemenet:	Kimenet:
8 2 1 7 4 9 2 9 1 2	22 1 1 1 2 1 2 2 1 2 2 2 1 3 1 3 2

2014. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Hofstadter (40 pont)

A Hofstadter R-S sorozat a következő szabályok szerint készül: R(1)=1, S(1)=2, R(n)=R(n-1)+S(n-1), S(n) a legkisebb 0-nál nagyobb egész szám, amely nem szerepel R(1)..R(n) és S(1)..S(n-1) között.

Készíts programot, amely előállítja a Hofstadter R-S sorozatban szereplő N. számpárt!

A bemenet egyetlen sorában N értéke van (1≤N≤100 000).

A kimenet egyetlen sorába két számot kell írni: R(N) és S(N) értékét!

Példa:

Bemenet (billentyűzet)		Kimenet (képernyő)	
#	Sortartalom [*magyarázat*]	#	Sortartalom [*magyarázat*]
1.	15 [*N értéke*]	1.	150 20 [*R(N) és S(N) értéke*]

2. feladat: Lottó (40 pont)

N lottóhúzásban N*5 számot húztak. Teljes sorozatnak nevezzük azt az egymás utáni legkevesebb húzásból álló sorozatot, amelyben az összes lehetséges lottószám előfordult legalább egyszer.

Készíts programot, amely megadja, hogy az N húzásban hány teljes sorozat van!

A bemenet első sorában a lottóhúzások száma van (18≤N≤100 000). A következő N sor az egyes lottóhúzások 5-5 kihúzott számát tartalmazza (mindegyik szám 1 és 90 közötti).

Megjegyzés: Egy teljes sorozathoz legalább 18 húzás kell (18*5=90), ha mindegyikben más számok fordulnak elő. Két teljes sorozathoz ezek alapján legalább 2*18 húzás kell, vagy másképp fogalmazva: a második teljes sorozathoz az első teljes sorozat után legalább újabb 18 húzás kell.

A kimenet egyetlen sorába a teljes sorozatok számát kell írni!

Példa:

Bemenet (billentyűzet)		Kimenet (képernyő)	
#	Sortartalom [*magyarázat*]	#	Sortartalom [*magyarázat*]
1.	3 [*a lottóhúzások száma*]	1.	0 [*a teljes sorozatok száma*]
2.	10 23 32 78 79 [*az 1. húzás számai*]		
3.	15 23 41 45 90 [*a 2. húzás számai*]		
4.	31 41 45 63 66 [*a 3. húzás számai*]		

3. feladat: Sorrend (70 pont)

Egy versenyen N versenyző indul, akiket 1 és N közötti sorszámukkal azonosítunk. A versenyzők sorrendjét kisorsoltuk, majd ugyanazt két hosszú papírszalagra felírtuk. Tréfás kedvű barátunk azonban mindkettőt elvágta ollóval néhány helyen (a két szalagot biztosan különböző helyeken), majd összekeverte

X X X	X	X X

Írj programot, amely a két összekevert sorrend alapján megadja a versenyzők eredeti sorrendjét!

A bemenet első sorában a versenyzők száma van (1≤N≤10 000). A második sorban az első papírszalag szétvágása után keletkezett darabok száma van (1≤E≤N). A következő E sorban egy-egy darab leírása található: a sor első száma a darabon levő számok száma, amelyet a darabon levő számok követnek. A következő sor a második szalag szétvágása után keletkezett darabok M számát tartalmazza (1≤M≤N), amelyet az egyes darabok leírását tartalmazó M sor követ.

A kimenet egyetlen sorába pontosan N számot kell írni: a versenyzők lehetséges eredeti sorrendjét!

Példa:

Bemenet (billentyűzet)		Kimenet (képernyő)	
#	Sortartalom [*magyarázat*]	#	Sortartalom [*magyarázat*]
1.	6 [*a versenyzők száma*]	1.	4 3 5 6 2 1 [*a jó sorrend*]
2.	3 [*az 1. szalag darabjai számai*]		
3.	3 4 3 5 [*az 1. szalag 1. darab hossza és számai*]		
4.	1 1 [*az 1. szalag 2. darab hossza és számai*]		
5.	2 6 2 [*az 1. szalag 3. darab hossza és számai*]		
6.	3 [*a 2. szalag darabjai számai*]		
7.	3 3 5 6 [*a 2. szalag 1. darab hossza és számai*]		
8.	2 2 1 [*a 2. szalag 2. darab hossza és számai*]		
9.	1 4 [*a 2. szalag 3. darab hossza és számai*]		

Kilencedik-tizedik osztályosok

1. feladat: Metró (22 pont)

Egy metróállomásra N időegységben érkeznek utasok, a K hosszú mozgólépcsőre legfeljebb ketten léphetnek egyszerre (azaz az érkezők közül ketten azonnal a mozgólépcső legfelső fokára kerülnek), a lépcsőn nincs mozgás – időegységenként mindenki egyet halad lefelé. A lépcső egy L utast befogadni képes váróterembe érkezik, az i-edik időegységben váróterembe lépőt ugyanabban az időegységben nem viheti el a metró. A metró M időegységenként jön és elviszi az összes várakozó utast. A beszállás 1 időegység alatt megtörténik. Kezdetben (a 0. időegységben) a lépcső és a váróterem is üres, az első metró az M. időegységben érkezik. Ha a váróterembe nem férnek be az utasok, akkor a metróállomást leállítják.

Készíts programot, amely megadja, hogy az egyes metrószerelvények hány utast visznek el! A végrehajtás vagy N+K+M időegység után fejeződjön be, vagy akkor, amikor a váróterem megtelik!

A bemenet első sorában az időegységek száma (1≤N≤100 000), a mozgólépcső hossza (1≤K≤100), a váróterem kapacitása (1≤L≤1000), a metrók követési távolsága (1≤M≤1000) és az érkező utasok száma (1≤U≤1 000 000) van. A következő U sor mindegyikében egy-egy utas érkezési ideje van ($0 \le Idő_i \le N$), nemcsökkenő sorrendben.

A kimenet első sorába az állomásról utasokat elvivő metrószerelvények S számát kell írni! A másodikba S szám kerüljön: az egyes metrószerelvények által elvitt utasok száma!

Példa:

Bemenet:	Kimenet:
12 4 10 8 12	3
3	2 9 1
3	
3	
3	
3	
3	
5	
6	
8	
8	
9	
12	

2. feladat: Párosítás (22 pont)

Két azonos létszámú csapat egyéni küzdelmet vív egymással, a te csapatod minden tagja az ellenfél egy másik tagjával mérkőzik párosításban. Két játékos közül mindig a nagyobb tudású győz. Ismerjük mindkét csapat tagjainak tudását.

Készíts programot, amely megad egy olyan párosítást, amely esetén a csapatod a lehető legtöbbször győz!

A bemenet első sora a csapattagok számát tartalmazza ($1 \leq N \leq 10\,000$). A második sor i-edik száma a csapatod i-edik tagjának erőssége ($1 \leq A_i \leq 100\,000$). A harmadik sor i-edik száma az ellenfél i-edik tagjának erőssége ($1 \leq B_i \leq 100\,000$).

A kimenet első sorába az elérhető győzelmek maximális számát kell írni! A második sor egy olyan párosítást adjon meg, amely maximális számú győzelmet eredményez csapatodnak! A sor pontosan N különböző egész számot tartalmazzon, az i-edik szám az ellenfél csapatából annak a versenyzőnek a sorszáma legyen, aki a csapatod i-edik tagjával mérkőzik! Több megoldás esetén bármelyik megadható.

Példa:

Bemenet	Kimenet
6	5
1 9 2 3 6 4	6 5 1 2 4 3
1 2 3 4 5 6	

3. feladat: Mingrel ábécé (32 pont)

A mingrel ábécé betűi szó szerint megegyeznek az angol ábécé betűivel (azaz a mingreleknek is pontosan 26 betűjük van), a betűk ábécé-sorrendje azonban eltér az angolétól.

Készíts programot, amely mingrel szavak ábécé sorrendje alapján megadja a mingrel ábécé betűinek lehetséges ábécé-sorrendjét!

A bemenet első sora az ismert mingrel szavak számát tartalmazza ($1 \leq N \leq 10\,000$), majd soronként egy-egy szót (legfeljebb 100 karakterből állhatnak) a mingrel ábécé szerinti sorrendben, csupa kisbetűvel írva.

A kimenet egyetlen sorába a mingrel ábécé betűinek egy lehetséges ábécé-sorrendjét kell írni (pontosan 26 betűt szóközök nélkül)! Több megoldás esetén bármelyik megadható.

Példa:

Bemenet	Kimenet
6 atom alma utt utas ember emberek	tauelbcdfghijkmnopqrsvwxyz

Megjegyzés: sok más jó megoldás is lehetséges.

4. feladat: Dobozok (32 pont)

Van N darab téglatest alakú dobozunk, mindegyiknek ismerjük a méreteit. A dobozokat egymásba szeretnénk pakolni, hogy minél kevesebb helyet foglaljanak. Egy dobozba olyan másik doboz tehető, amelynek mindhárom mérete kisebb az adott doboz méreteinél, de a dobozok tetszőlegesen forgathatók (azaz pl. egy (7,5,3) méretű dobozba betehető egy (4,2,6) méretű doboz).

Készíts programot, amely megadja a legtöbb dobozból álló dobozsorozatot, amelyek egymásba pakolhatók!

A bemenet első sorában a dobozok száma ($1<N\leq 2000$) van. A következő N sor mindegyike 3 pozitív egész számot tartalmaz, az egyes dobozok méretét ($1\leq x_i, y_i, z_i\leq 10\,000$).

A kimenet első sorába a legtöbb dobozból álló dobozsorozat H elemszámát kell írni, amelyek egymásba pakolhatók! A következő sorba pontosan H számot kell írni, a leghosszabb ilyen dobozsorozatban szereplő dobozok sorszámait! Az első legyen közülük a legnagyobb doboz, s minden dobozt olyan kövessen, amely az adott dobozba belefér! Több megoldás esetén bármelyik megadható.

Példa:

Bemenet	Kimenet
6 2 3 8 1 4 2 4 2 7 2 1 3 5 3 9 1 2 3	3 5 3 2

5. feladat: Szállítás (42 pont)

Egy nagyvállalat N városban üzemeltet egy-egy gyárat. Minden gyárban van raktára, ahol termékeket tárol. Minden gyárból, ahol több terméket gyártottak, mint az ottani raktár kapacitása, el kell szállítani másik raktárba. A szállítást közúton végzik kamionokkal, minden kamion legfeljebb K darab terméket tud szállítani. A közlekedési hatóság korlátozása miatt minden városból csak egy megadott másik városba mehet közvetlenül kamion. Van egy központi város, amelyikbe bármely másik városból el lehet jutni, itt van a központi raktár. Ha egy kamion az útja során áthalad egy városon, akkor az ottani raktárba lerakhat termékeket, ha ott van még hely, illetve felvehet elszállítandó terméket, ha van hely rajta. Add meg, hogy minimum hány kamion kell az összes termék elszállításához és mekkora további raktárt kell építeni a központi városba, hogy minden termék raktárba kerüljön!

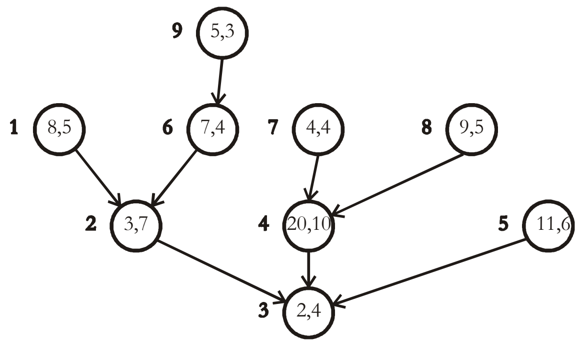
Készíts programot, amely megválaszolja, hogy minimum hány kamion kell az összes termék elszállításához és mekkora raktárt kell építeni a központi városba, hogy minden termék raktárba kerüljön!

A bemenet első sorában a városok száma (1≤N≤1000), és a szállító kamionok kapacitása (1≤K≤1000) van. A második sorban az egyes városokban termelt termékek száma van (legfeljebb 2000). A harmadik sorban az egyes városokban tárolható termékek száma van (legfeljebb 3000). A negyedik sorban az i-edik szám annak a városnak a sorszáma, ahova az i-edik városból közvetlenül mehet kamion. A sorban egy helyen szerepel 0, ha ez az i-edik szám a sorban, akkor ez azt jelenti, hogy az i-edik városban van a központi raktár.

A kimenet első sorába a minimálisan szükséges kamionok számát kell írni, a második sorba pedig az építendő központi raktár méretét!

Példa:

Bemenet	Kimenet
9 10	5
8 3 2 20 11 7 4 9 5	21
5 7 4 10 6 4 4 5 3	
2 3 0 3 3 2 4 4 6	



Tizenegyedik-tizenharmadik osztályosok

1. feladat: Metró (20 pont)

Egy metróállomásra N időegységben érkeznek utasok, a K hosszú mozgólépcsőre legfeljebb ketten léphetnek egyszerre (azaz az érkezők közül ketten azonnal a mozgólépcső legfelső fokára kerülnek), a lépcsőn nincs mozgás – időegységenként mindenki egyet halad lefelé. A lépcső egy L utast befogadni képes váróterembe érkezik, az i-edik időegységben váróterembe lépőt ugyanabban az időegységben nem viheti el a metró. A metró M időegységenként jön, kiszáll belőle adott számú utas, és elviszi az összes metróra várakozó utast. A ki- és beszállás 1 időegység alatt megtörténik. A felfelé menő mozgólépcsőre várakozó utasok közül egy időegységben legfeljebb 2 léphet a lépcsőre. Aki most szállt le a metróról, az leghamarabb a következő időegységben léphet a mozgólépcsőre. Kezdetben (a 0. időegységben) a lépcső és a váróterem is üres, az első metró az M. időegységben érkezik. Ha a váróterembe nem férnek be az utasok, akkor a metróállomás működését leállítják.

Készíts programot, amely megadja, hogy az egyes metrószerelvények hány utast visznek el! A végrehajtás vagy N+K+M időegység után fejeződjön be, vagy amikor a metróállomás működését leállítják!

A bemenet első sorában az időegységek száma (1≤N≤1 000 000), a mozgólépcső hossza (1≤K≤100), a váróterem kapacitása (1≤L≤1000), a metrók követési távolsága (1≤M≤1000) és az érkező utasok száma (1≤U≤1 000 000) van. A következő U sor mindegyikében egy-egy utas érkezési ideje van ($0 \le Idő_i \le N$), nemcsökkenő sorrendben. A következő sorban az egyes metrószerelvényekről leszállók száma van, a szerelvények érkezési sorrendjében.

A kimenet első sorába az állomásról utasokat elvivő metrószerelvények S számát kell írni! A másodikba S szám kerüljön: az egyes metrószerelvények által elvitt utasok száma!

Példa:

Bemenet:

```
12 4 10 8 12
3
3
3
3
3
3
5
6
8
8
9
12
3 5 2
```

Kimenet:

```
3
2 9 1
```

2. feladat: Fénykép (30 pont)

Egy rendezvényre vendégek érkeznek. Ismerjük mindenkinek az érkezési és távozási időpontját. A szervező megbízott egy fényképészt, hogy a résztvevőkről csoportképeket készítsen. A fényképész minél hamarabb szeretne végezni, ezért amint jelen van legalább K vendég, akkor közülük pontosan K vendéget lefényképez egy csoportképen, azaz csak abban dönthet, hogy adott időpontban kiket fényképez le. Egy időpontban csak egy fényképet tud készíteni, és minden vendég legfeljebb 1 képen szerepelhet. A vendégek már az érkezési időpontjukban lefényképezhetők és az utolsó lehetőség a lefényképezésükre a távozási időpontjuk.

Készíts programot, amely megadja, hogy maximum hány fényképet tud készíteni a fényképész, és megadja, hogy az egyes képeken kik lesznek!

A bemenet első sorában a vendégek száma ($1\le N\le 100\,000$), valamint a K értéke ($1\le K\le 100$) van. A következő N sor mindegyikében egy-egy vendég érkezési és távozási időpontja ($1\le E_i<T_i\le 10\,000$) van, érkezési időpont szerint nemcsökkenő sorrendben.

A kimenet első sorába a fényképezések maximális F számát kell írni! A következő F sor mindegyike pontosan K különböző egész számot tartalmazzon, azon vendégek sorszámait, akit az adott időpontban a csoportképen lesznek! Több megoldás esetén bármelyik megadható.

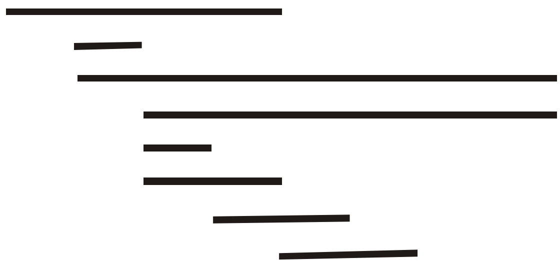
Példa:

Bemenet:

```
8 3
1 5
2 3
2 9
3 9
3 4
3 5
4 6
5 7
```

Kimenet:

```
2
2 1 3
5 6 4
```



3. feladat: Koncert (30 pont)

A nagy érdeklődéssel várt koncertre jegyet lehet igényelni. A szervező célja, hogy a lehető legtöbb ülőhelyet adja el úgy, hogy az eladott jegy az igénylőnek megfelelő legyen.

Készíts programot, amely megadja, hogy mekkora az elérhető legnagyobb bevétel, és mely igények teljesítése adja az elérhető legnagyobb bevételt!

A bemenet első sorában az ülőhelyek száma (1≤M≤1000), és az igények száma (1≤N≤2000) van. A következő N sor mindegyike egy-egy igényt tartalmaz – az első H hely közül K egymás mellettit igényel (K≤H≤M).

A kimenet első sorába a legtöbb eladható ülőhely számát és a kielégített igények S számát kell írni. A következő S sor mindegyike egy kielégített igénylő sorszámot és az első ülőhely sorszámát tartalmazza, amit az igénylő kap! Több megoldás esetén bármelyik megadható.

Példa:

Bemenet:	Kimenet:
10 7	9 4
2 1	7 6
3 2	6 4
8 2	2 2
8 5	1 1
7 3	
6 2	
9 4	

4. feladat: Robot (30 pont)

Egy négyzetrácsos elrendezésű raktárban robot alkalmazásával dolgoznak. A raktár egy mezőjét a négyzetrácsos elrendezésben a mező (x,y) koordinátájával azonosítják, ahol x a sor, y pedig az oszlop koordinátája, a sorokat alulról felfelé, az oszlopokat balról jobbra sorszámozzuk, a bal alsó mező koordinátái (1,1). A robot egy lépésben a szomszédos mezőre léphet vagy felfelé, vagy jobbra. A raktár N mezőjén van tárolva áru. A robotnak az (1,1) mezőről indulva, legfeljebb L lépés megtételével olyan útvonalon kell haladnia, amelyen a lehető legtöbb árut tartalmazó mező van.

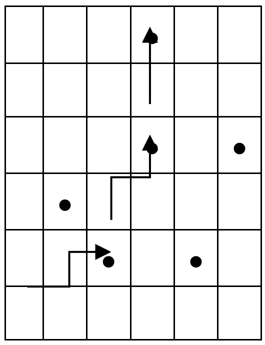
Készíts programot, amely megoldja a feladatot!

A bemenet első sorában az árut tartalmazó mezők száma (1≤N≤10 000) és a robot által megtehető lépések maximuma (1≤L≤2 000 000). A további N sor mindegyikében egy-egy olyan mező koordinátái vannak, ahol áru van (legfeljebb 2 000 000).

A kimenet első sorába azt a legnagyobb M számot kell írni, ahány árut tartalmazó mezőn áthaladhat a robot! A második sorba pontosan M számot kell írni, a bejárás sorrendjében a mezők bemenetbeli sorszámait! Több megoldás esetén bármelyik megadható.

Példa:

Bemenet:	Kimenet:
6 8	3
3 2	1 4 6
2 3	
5 2	
4 4	
6 4	
4 6	



5. feladat: Szerviz (40 pont)

Egy számítógépes hálózat N csomópontot tartalmaz, bizonyos csomópont párokat kétirányú kommunikációt biztosító közvetlen átviteli vonal köt össze. A hálózat összefüggő, tehát bármely két csomópont között lehet átvitel, esetleg több közbülső vonalon keresztül. A hálózat üzemeltetése kétféle, A és B szolgáltatást biztosító szoftvert akar telepíteni bizonyos csomópontokba. A telepítésnél az következő feltételeket kell teljesíteni:

1. Minden csomópontba legfeljebb egy szolgáltatás telepíthető,
2. N/3 csomópontba nem telepíthető egyik szolgáltatás sem,
3. bármely P csomóponthoz legyen olyan Q csomópont, hogy Q-ba van telepítve az A szolgáltatás és Q=P, vagy Q legfeljebb egy közbülső csomóponton keresztül elérhető P-ből,
4. bármely P csomóponthoz legyen olyan Q csomópont, hogy Q-ba van telepítve a B szolgáltatás és Q=P, vagy Q legfeljebb egy közbülső csomóponton keresztül elérhető P-ből.

Készíts programot, amely megadja, hogy mely csomópontokba kell A, illetve B szolgáltatást telepíteni!

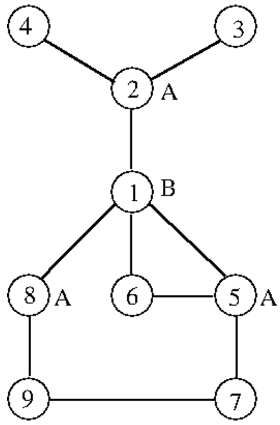
A bemenet első sorában a csomópontok száma (3≤N≤1000) és a közvetlen vonalak száma (2≤M≤100 000) van. A további M sor mindegyike két csomópont sorszámát tartalmazza, amelyek között van közvetlen átviteli vonal. Bármely számpár legfeljebb egyszer szerepel a bemenetben.

A kimenet első sor azon csomópontokat adja meg, amelyekbe az A szolgáltatást telepítjük; az első szám ezeknek a csomópontoknak a K1 száma legyen, a további K1 szám pedig a csomópontok sorszámai (tetszőleges sorrendben)! Hasonlóan, a második sor azon csomópontokat adja meg, amelyekbe a B szolgáltatást telepítjük; az első szám ezeknek a csomópontoknak a K2 száma legyen, a további K2 szám pedig a csomópontok sorszámai (tetszőleges sorrendben)! Több megoldás esetén bármelyik megadható.

Példa:

```
Bemenet:
9 10
3 2
2 1
2 4
1 5
1 6
5 6
5 7
8 1
8 9
7 9
```

```
Kimenet:
3 2 5 8
1 1
```



2014. A verseny végeredménye

I. korcsoport

1	Gáspár Attila	Földes Ferenc Gimnázium, Miskolc
2	Kurucz György	Veres Péter Gimnázium, Budapest
3	Jakab Farkas Attila	Fazekas Mihály Gimnázium, Debrecen
4	Molnár-Sáska Zoltán	Fazekas Mihály Gimnázium, Budapest
5	Kiss Gergely	Fazekas Mihály Gimnázium, Budapest
6	Németh Balázs	Lehel Vezér Gimnázium, Jászberény
7	Alexy Marcell	Fazekas Mihály Gimnázium, Budapest
8	Szemerédi Levente	Radnóti Miklós Gimnázium, Szeged
9	Fuisz Gábor	Veres Péter Gimnázium, Budapest
10	Székely András	Református Gimnázium, Szentendre

II. korcsoport

1	Hornák Bence	Berzsenyi Dániel Gimnázium, Budapest
2	Mernyei Péter	Radnóti Miklós Gimnázium, Budapest
3	Kovács Benedek	Fazekas Mihály Gimnázium, Budapest
4	Zarándy Álmos	Fazekas Mihály Gimnázium, Budapest
5	Dobos-Kovács Mihály	Szent István Gimnázium, Budapest
6	Zalavári Márton	Zrínyi Miklós Gimnázium, Zalaegerszeg
7	Kovács Tamás	Kölcsey Ferenc Főgimnázium, Szatmárnémeti
8	Radnai László	Veres Péter Gimnázium, Budapest
9	Bencze Gábor Péter	Boronkay György Műszaki Szakközépiskola, Vác
10	Almási Nóra	Fazekas Mihály Gimnázium, Debrecen

III. korcsoport

1	Nagy Vendel	Fazekas Mihály Gimnázium, Debrecen
2	Weisz Ambrus	Fazekas Mihály Gimnázium, Budapest
3	Székely Szilveszter	Neumann János Középiskola és Kollégium, Eger
4	Végvári Zalán	Fazekas Mihály Gimnázium, Debrecen
5	Fehér Balázs	Berzsenyi Dániel Gimnázium, Budapest
6	Leitereg Miklós	Veres Péter Gimnázium, Budapest
7	Almási Péter Béla	Fazekas Mihály Gimnázium, Debrecen
8	Matkovics Gábor	Földes Ferenc Gimnázium, Miskolc
9	Kemenes Balázs	Szent István Gimnázium, Budapest
10	Somogyvári Kristóf	Ságvári Endre Gimnázium, Szeged

II.

Megoldások, értékelések

Nemes Tihamér
Nemzetközi Informatikai Tanulmányi Verseny

2010. Első forduló

Ötödik-nyolcadik osztályosok

Számítógép nélküli feladatok

<u>1. feladat</u>: Számok (18 pont)

A. II B. IV C. XIII D. XXV E. XXXVII F. XIX

G. XL H. XLI I. XLIX

Helyes megoldásonként 2-2 pont adható összesen 18 pont

<u>2. feladat</u>: Testvérek (27 pont)

A. A szegélyezettek a helyes utasítások. Ha a versenyző ugyanezeket a hibákat másképpen vette észre, arra is jár pont.

`Ha A(i,`$\boxed{2}$`)=X akkor DB:=DB+1; SX(DB):=A(i,`$\boxed{1}$`)` 2+2 pont

`i:=1; DB:=`$\boxed{0}$ és ekkor `SY(DB):=A(i,1); DB:=DB+1` sorrendje felcserélendő vagy `Ciklus amíg i≤N és DB`$\boxed{\leq}$`2` 3 pont

`Ha A(i,`$\boxed{2}$`)=Y akkor` 3 pont

`Ciklus amíg i`$\boxed{\leq}$`N és DB`$\boxed{\leq}$`2` 2+2 pont

Testvérek feltételeiből hiányzik: `vagy SX(1)=SY(2) és SX(2)=SY(1)` 4 pont

Féltestvérek feltételeiből hiányzik: `vagy SX(1)=SY(2) vagy SX(2)=SY(1)` 4 pont

B. `DB<2` a helyes feltétel, amire a ciklus hamarabb befejeződhet 5 pont

<u>3. feladat</u>: Járda (25 pont)

A. 1,2,3 2+2+2 pont

B. 2+2+2 pont

C. 8 3 pont

D. 13 4 pont

E. 89 6 pont

Számítógépes feladat – VÁLASZTHATÓ

<u>4. feladat</u>: Eső (30 pont)

A feladat három programozási tétel alkalmazásáról szól:

```
Összegzés(N,Eső,A):
  A:=0
  Ciklus i=1-től N-ig
    A:=A+Eső[i]
  Ciklus vége
Eljárás vége.
```

```
Maximum(N,Eső,B):
  B:=1
  Ciklus i=2-től N-ig
    Ha Eső[i]>Eső[B] akkor B:=i
  Ciklus vége
Eljárás vége.
```

```
Darabszám(N,Eső,C):
  C:=0
  Ciklus i=1-től N-ig
    Ha Eső[i]>0 akkor C:=C+1
  Ciklus vége
Eljárás vége.
```

Számítógép nélküli feladat – VÁLASZTHATÓ

4. feladat: Kannák (30 pont)

Figyelem: Lehet más helyes megoldás is! Legkevesebb elemszámú megoldások:

A. AEDE B. BFBFBF C. AEDEDEAE D. AE E. BFBF

Helyes műveletsorra esetenként 3-3 pont adható összesen 15 pont

Legkisebb elemszámú műveletsorra esetenként további 3-3 pont adható összesen 15 pont

Kilencedik-tizedik osztályosok

1. feladat: Számok (20 pont)

Helyes megoldásonként A..D-re 1-1, a többire 2-2 pont adható összesen 20 pont

A. II B. IV C. XIII D. XXV E. XCIX F. XIX

G. XL H. XLI I. XLIX J. LXXVII K. LXIV L. XLIV

2. feladat: Testvérek (24 pont)

A. A szegélyezettek a helyes utasítások. Ha a versenyző ugyanezeket a hibákat másképpen vette észre, arra is jár pont.

`Ciklus amíg i`$\boxed{\le}$`N és DB<2` 2 pont

`Ha `$\boxed{\texttt{DB>0 és}}$` A(i,1)=SX(1) vagy `$\boxed{\texttt{DB>1 és}}$` A(i,1)=SX(2)` 3+3 pont
(ez az elvi hibák között is lehet)

`akkor D:=D+1; T(`$\boxed{\texttt{D}}$`):=A(`$\boxed{\texttt{i}}$`,2)` 3+3 pont

B. Saját magát is megadja testvérként 5 pont

Az édestestvéreket kétszer adja meg 5 pont

3. feladat: Mit csinál? (21 pont)

A. Első: D=4, E=(Alfa,Béta,Gamma,Delta) 1+1 pont

Második: D=4, E=(Alfa,Béta,Gamma,Delta), F=(1,1,1,6) 1+1+2 pont

Harmadik: H=Alfa 3 pont

B. Első: Azokat a játékosokat adja meg, akik legalább egyszer bejelentkeztek az N nap alatt 4 pont

Második: Minden bejelentkezett játékosra megadja, hogy mennyi volt a minimális napi belépései száma 4 pont

Harmadik: Megadja a legtöbb napon belépett játékost 4 pont

4. feladat: Kannák (15 pont)

Figyelem: Lehet más helyes megoldás is! Lehetséges legkevesebb elemszámú megoldások:

A. AEDE, B. BFBFBF, C. AEDEDEAE D. AEDEDEAEDE E. BFBFBFCFBF

Helyes műveletsorra esetenként 2-2 pont adható összesen 10 pont

Legkisebb elemszámú műveletsorra esetenként további 1-1 pont adható összesen 5 pont

5. feladat: Járda (20 pont)

A. 1,2,4 1+1+1 pont

B. Helyes rajzonként 1-1 pont összesen 4 pont

C. 13 3 pont

D. 24 4 pont

E. 274 6 pont

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Számok (20 pont)

Helyes megoldásonként A..D-re 1-1, a többire 2-2 pont adható összesen 20 pont

A. II	B. XIV	C. XLIII	D. LXXXV	E. XCIX	F. CXXIII
G. CCCXL	H. CDLI	I. CDXCIX	J. DCCCLXXVII	K. DCXIX	L. CMXCIX

2. feladat: Testvérek (22 pont)

A. A szegélyezettek a helyes utasítások. Ha a versenyző ugyanezeket a hibákat másképpen vette észre, arra is jár pont.

Ciklus amíg i$\boxed{\le}$N 1 pont

Ha A(i,$\boxed{1}$)=X akkor DB:=DB+1; SX(DB):=A($\boxed{i}$,$\boxed{2}$) 1+1+2 pont

Ciklus amíg j≤DB és A(i,2)$\boxed{\ne}$SX($\boxed{j}$) 2+2 pont

Ha j$\boxed{\le}$DB akkor ... 1 pont

T($\boxed{D}$):=A(i,1) 2 pont

B. Saját magát is megadja 5 pont

Akivel több gyereke is van, azokat többször megadja 5 pont

3. feladat: Mit csinál? (22 pont)

A. Egy: D=2, E=(Alfa,Gamma) 1+1 pont

Második: D=4, E=(Alfa,Béta,Gamma,Delta), F=(4,15,1,6) 1+1+2 pont

Harmadik: H=Béta 4 pont

B. Első: Azokat a játékosokat adja meg, akik minden nap bejelentkeztek az N nap alatt 4 pont

Második: Minden bejelentkezett játékosra megadja, hogy mennyi volt a maximális napi belépései száma 4 pont

Harmadik: Megadja a legtöbbször belépett játékost 4 pont

4. feladat: Kannák (15 pont)

Figyelem: Lehet más helyes megoldás is! Legkevesebb elemszámú megoldások:

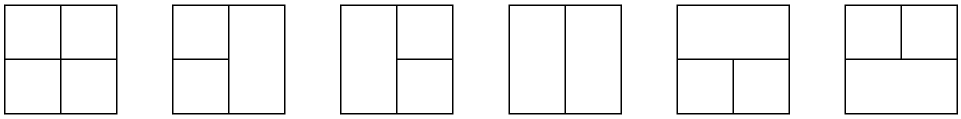
A. AEDE B. BFBFBF C. AEDEDEAE D. AEDEDEAEDE E. BFBFBFCFBF

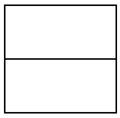
Helyes műveletsorra esetenként 2-2 pont adható összesen 10 pont

Legkisebb elemszámú műveletsorra esetenként további 1-1 pont adható összesen 5 pont

5. feladat: Járdakövezés (21 pont)

A 3 pont





2 pont adható, ha legalább 5 ábra jó.
1 pont adható, ha legalább 3 ábra jó.

B. 22 3 pont

C. 71 3 pont

D. 228 3 pont

E. 2356 4 pont

F. 7573 5 pont

2010. Második forduló

Ötödik-nyolcadik osztályosok

1. feladat: Kártya (25 pont)

A megolodáshoz három konstans tömböt használunk, az egyes kártyalapok elnevezéseiről és pontértékéről:

```
szín=('piros','zöld','tök','makk')

figura=('7-es','8-as','9-es','10-es',
        'alsó','felső','király','ász')

érték=(7,8,9,10,2,3,4,11)
```

Meg kell határoznunk az adott lapok figuráját, amihez tartozó értéket kell felhasználnunk az összérték kiszámításához, piros kártya esetén a dupláját.

```
Pontok(N,S,F,E):
  E:=0
  Ciklus i=1-től n-ig
    j:=1
    Ciklus amíg S[i]≠szín[j]
      j:=j+1
    Ciklus vége
    Ha j=1 akkor sz:=2 különben sz:=1
    j:=1
    Ciklus amíg F[i]≠figura[j]
      j:=j+1
    Ciklus vége
    E:=E+sz*érték[j]
  Ciklus vége
Eljárás vége.
```

2. feladat: Bábu (25 pont)

A legegyszerűbb megoldás egy szimuláció: kövessük a bábuk mozgását időegységenként! Ha ugyanarra a helyre érnének, vagy átlépnének egymásokon, akkor a szimuláció megáll, megtörtént az első ütközés. Ha elértük az időkorlátot (K), akkor is megállunk. Minden bábut a helyével és haladási irányával követünk.

A szimuláció a párhuzamosság (a bábuk egyszerre lépnek) feloldásáról szól. Tároljuk a `tábla` tömbben a játéktábla aktuális állását (melyik bábu melyik pozíción van az aktuális időegységben), az `új` tömbben pedig a következő állást (melyik bábu melyik pozícióban lesz a következő időegységben)!

```
Első ütközés(N,Db,Bábu,K,Ütközés,Idő):
  tábla:=(0,…,0)
  Ciklus i=1-től Db-ig
    tábla[Bábu[i].hely]:=i
  Ciklus vége
  Idő:=0; Ütközés:=hamis; új:=tábla
  Ciklus amíg Idő≤K és nem Ütközés
    Idő:=Idő+1
    Ciklus i=1-től Db-ig
      Ha Bábu[i].hely=1 és Bábu[i].irány='B'
         akkor Bábu[i].irány:='J'
      Ha Bábu[i].hely=N és Bábu[i].irány='J'
         akkor Bábu[i].irány:='B'
      Ha Bábu[i].irány='J'
         akkor Ha új[Bábu[i].hely+1]>0 vagy
                  tábla[Bábu[i].hely+1]<tábla[Bábu[i].hely]) és
                  Bábu[tábla[Bábu[i].hely+1]].irány='B'
                     akkor Ütközés:=igaz
               új[Bábu[i].hely]:=0; új[Bábu[i].hely+1]:=i
               Bábu[i].hely:=Bábu[i].hely+1
      különben ha Bábu[i].irány='B'
         akkor Ha új[Bábu[i].hely-1]>0 vagy
                  tábla[Bábu[i].hely-1]<tábla[Bábu[i].hely] és
                  Bábu[tábla[Bábu[i].hely-1]].irány='J'
                     akkor Ütközés:=igaz
               új[Bábu[i].hely]:=0; új[Bábu[i].hely-1]:=i
               Bábu[i].hely:=Bábu[i].hely-1
      tábla:=új
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

3. feladat: Járdakövezés (25 pont)

A járdalefedést 1 vagy 2 méretű lappal indíthatjuk. Ha 1-essel indítjuk, akkor marad egy N-1 hosszúságú járda:

Ha 2-essel indítunk, akkor N-2 hosszúságú járda marad:

Azaz az N hosszúságú járda lefedései száma az N-1 és N-2 hosszúságú járdák lefedései számának összege, amik éppen a Fibonacci számok:

```
Lefedésszám(N,Db):
  A[1]:=1; A[2]:=2
  Ciklus i=3-tól N-ig
    A[i]:=A[i-2]+A[i-1]
  Ciklus vége
  Db:=A[N]
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Kép (20 pont)

Meg kell keresni a legelső eltérő pontot a két mátrixban, 4 különböző irányból: legbaloldalibb oszlop, legalsó sor, legjobboldalibb oszlop, legfelső sor!

```
Levág(N,M,T,U,Bal,Lent,Jobb,Fent):
  i:=1; j:=1
  Ciklus amíg i≤N és t[i,j]=u[i,j])
    Ha j<M akkor j:=j+1 különben i:(i+1; j:=1
  Ciklus vége
  Fent:=i-1
  i:=1; j:=1
  Ciklus amíg j≤M és t[i,j]=u[i,j]
    Ha i<N akkor i:=i+1 különben j:=j+1; i:=1
  Ciklus vége
  Bal:=j-1
  i:=N; j:=M
  Ciklus amíg i≥1 és t[i,j]=u[i,j]
    Ha j>1 akkor j:=j-1 különben i:=i-1; j:=M
  Ciklus vége
  Lent:=N-i
  i:=N; j:=M
  Ciklus amíg j≥1 és t[i,j]=u[i,j]
    Ha i>1 akkor i:=i-1 különben j:=j-1; i:=N
  Ciklus vége
  Jobb:=M-j
Eljárás vége.
```

2. feladat: Játék (20 pont)

A feladat az előző korcsoportos feladat nehezítése, két dimenzióban. Ugyanúgy szimulációs megoldást készítünk rá, tárolva a bábuk sor- és oszlopindexét, valamint irányát. A játéktáblát itt is két időegységben ábrázoljuk (`tábla`, `új`).

```
Ütközés kereső(N,M,Db,Bábu,Ütközés,Idő):
  tábla:=(0,…,0)
  Ciklus i=1-től Db-ig
    tábla[Bábu[i].sor,Bábu[i].oszlop]:=i
  Ciklus vége
  Idő:=0; Ütközés:=hamis; új:=tábla
  Ciklus amíg Idő≤K és nem Ütközés
    Idő:=Idő+1
    Ciklus i=1-től Db-ig
      Ha Bábu[i].sor=1 és Bábu[i].irány='F'
         akkor Bábu[i].irány:=' '
      különben ha Bábu[i].sor=N és Bábu[i].irány='L'
         akkor Bábu[i].Irány:=' '
      különben ha Bábu[i].oszlop=1 és Bábu[i].irány='B'
         akkor Bábu[i].irány:=' '
      különben ha Bábu[i].oszlop=M és Bábu[i].irány='J'
         akkor Bábu[i].irány:=' '
      Ha Bábu[i].irány='F' akkor Felfelé
      Ha Bábu[i].irány='L' akkor Lefelé
      Ha Bábu[i].irány='J' akkor Jobbra
      Ha Bábu[i].irány='B' akkor Balra
      tábla:=új
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Ütközik, ha olyan helyre lépne, ahova már lépett valaki, vagy vele szembe haladót átlépne:

```
Felfelé:
  Ha új[Bábu[i].sor-1,Bábu[i].oszlop]>0 vagy
     tábla[Bábu[i].sor-1,Bábu[i].oszlop]<
                        tábla[Bábu[i].sor,Bábu[i].oszlop] és
     Bábu[tábla[Bábu[i].sor-1,Bábu[i].oszlop]].irány='L'
          akkor Ütközés:=igaz
  új[Bábu[i].sor Bábu[i].oszlop]:=0
  új[Bábu[i].sor-1,Bábu[i].oszlop]:=i
  Bábu[i].sor:= Bábu[i].sor-1
Eljárás vége.
```

```
Lefelé:
  Ha új[Bábu[i].sor+1,Bábu[i].oszlop]>0 vagy
     tábla[Bábu[i].sor+1,Bábu[i].oszlop]<
                      tábla[Bábu[i].sor,Bábu[i].oszlop] és
     Bábu[tábla[Bábu[i].sor+1,Bábu[i].oszlop]].irány='F'
          akkor Ütközés:=igaz
  új[Bábu[i].sor,Bábu[i].oszlop]:=0
  új[Bábu[i].sor+1,Bábu[i].oszlop]:=i
  Bábu[i].sor:= Bábu[i].sor+1
Eljárás vége.
```

```
Jobbra:
  Ha új[Bábu[i].sor,Bábu[i].oszlop+1]>0 vagy
     tábla[Bábu[i].sor,Bábu[i].oszlop+1]<
                      tábla[Bábu[i].sor,Bábu[i].oszlop] és
     Bábu[tábla[Bábu[i].sor,Bábu[i].oszlop+1]].irány='B'
          akkor Ütközés:=igaz
  új[Bábu[i].sor,Bábu[i].oszlop]:=0
  új[Bábu[i].s,Bábu[i].oszlop+1]:=i
  Bábu[i].oszlop:=Bábu[i].oszlop+1
Eljárás vége.
```

```
Balra:
  Ha új[Bábu[i].sor,Bábu[i].oszlop-1]>0 vagy
     tábla[Bábu[i].sor,Bábu[i].oszlop-1]<
                      tábla[Bábu[i].sor,Bábu[i].oszlop]és
     Bábu[tábla[Bábu[i].sor,Bábu[i].oszlop-1]].irány='J'
          akkor Ütközés:=igaz
  új[Bábu[i].sor,Bábu[i].oszlop]:=0
  új[Bábu[i].sor,Bábu[i].oszlop-1]:=i
  Bábu[i].oszlop:=Bábu[i].oszlop-1
Eljárás vége.
```

3. feladat: Ütemezés (20 pont)

Biztosan kell annyi idő a gyártásra, amennyi ideig az A műveletet el tudja végezni az összes alkatrészen, illetve ugyanez a B műveletre is. Egyszerű esetben a megoldási idő e két időtartam maximuma.

Akkor nem jó ez a megoldás, ha nem oszthatók be olyan sorrendbe az alkatrészek, hogy a két műveletet mindig különböző alkatrészen kelljen elvégezni. Ez akkor fordulhat elő, hogy ha van olyan alkatrész, amely két műveletének összideje nagyobb vagy az A művelet, vagy a B művelet összidejénél.

```
Ütemezés(N,a,b,Ered):
  at:=0; bt:=0
  Ciklus i=1-től N-ig
    at:=at+a[i]; bt:=bt+b[i]
  Ciklus vége
  i:=1
  Ciklus amíg i≤N és (at>a[i]+b[i] vagy bt>a[i]+b[i])
    i:=i+1
  Ciklus vége
  Ha i≤N akkor Ered:=a[i]+b[i]
  különben Ha at>bt akkor Ered:=at
  különben Ered:=bt
Eljárás vége.
```

<u>4. feladat</u>: Járdakövezés (15 pont)

A járdalefedést 1, 2 vagy 3 méretű lappal indíthatjuk. Ha 1-essel indítjuk, akkor marad egy N-1 hosszúságú járda:

Ha 2-essel indítunk, akkor N-2 hosszúságú járda marad:

Ha 3-assal indítunk, akkor N-3 hosszúságú járda marad:

Azaz az N hosszúságú járda lefedései száma az N-1, N-2 és N-3 hosszúságú járdák lefedései számának összege, amik Fibonacci szám variációk:

```
Lefedésszám(N,Db):
  A[1]:=1; A[2]:=2; A[3]:=4
  Ciklus i=4-től N-ig
    A[i]:= A[i-3]+A[i-2]+A[i-1]
  Ciklus vége
  Db:=A[N]
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

<u>1. feladat</u>: Műhold (15 pont)

Szinte azonos a feladat az előző korcsoport hasonló feladatával, csak az eredmény megadása más.

Meg kell keresni a legelső eltérő pontot a két mátrixban, 4 különböző irányból: legbaloldalibb oszlop, legalsó sor, legjobboldalibb oszlop, legfelső sor!

```
Levág(N,M,T,U,Bal,Lent,Jobb,Fent):
  i:=1; j:=1
  Ciklus amíg i≤N és t[i,j]=u[i,j])
    Ha j<M akkor j:=j+1 különben i:(i+1; j:=1
  Ciklus vége
  Fent:=i
  i:=1; j:=1
  Ciklus amíg j≤M és t[i,j]=u[i,j]
    Ha i<N akkor i:=i+1 különben j:=j+1; i:=1
  Ciklus vége
  Bal:=j
```

```
  i:=N; j:=M
  Ciklus amíg i≥1 és t[i,j]=u[i,j]
    Ha j>1 akkor j:=j-1 különben i:=i-1; j:=M
  Ciklus vége
  Lent:=i
  i:=N; j:=M
  Ciklus amíg j≥1 és t[i,j]=u[i,j]
    Ha i>1 akkor i:=i-1 különben j:=j-1; i:=N
  Ciklus vége
  Jobb:=j
Eljárás vége.
```

2. feladat: Játék (15 pont)

Ez a feladat is szinte azonos az előző korcsoport hasonló feladatával, itt a tábla szélén a bábuknak vissza kell fordulni.

Szimulációs megoldást készítünk rá, tárolva a bábuk sor- és oszlopindexét, valamint irányát. A játéktáblát itt is két időegységben ábrázoljuk (`tábla`, `új`).

```
Ütközés kereső(N,M,Db,Bábu,Ütközés,Idő):
  tábla:=(0,…,0)
  Ciklus i=1-től Db-ig
    tábla[Bábu[i].sor,Bábu[i].oszlop]:=i
  Ciklus vége
  Idő:=0; Ütközés:=hamis; új:=tábla
  Ciklus amíg Idő≤K és nem Ütközés
    Idő:=Idő+1
    Ciklus i=1-től Db-ig
      Ha Bábu[i].sor=1 és Bábu[i].irány='F'
         akkor Bábu[i].irány:='L'
      különben ha Bábu[i].sor=N és Bábu[i].irány='L'
         akkor Bábu[i].Irány:='F'
      különben ha Bábu[i].oszlop=1 és Bábu[i].irány='B'
         akkor Bábu[i].irány:='J'
      különben ha Bábu[i].oszlop=M és Bábu[i].irány='J'
         akkor Bábu[i].irány:='B'
      Ha Bábu[i].irány='F' akkor Felfelé
      Ha Bábu[i].irány='L' akkor Lefelé
      Ha Bábu[i].irány='J' akkor Jobbra
      Ha Bábu[i].irány='B' akkor Balra
      tábla:=új
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

A négy mozgás azonos az előző korcsoportban leírtakkal.

3. feladat: Futár (15 pont)

Első lépésként a futárok (azaz a gráf élei) leírásából elő kell állítani a gráfot, majd ebben a gráfban kell élhossz összegek szerinti legrövidebb utak hosszát számolni. (A kis korlátok miatt a gráfot akár csúcsmátrix-szal is ábrázolhatjuk.)

```
Futár(N,M,Fut,min):
  cs:=(+∞, …, +∞); tav:=(+∞, …, +∞)
  Ciklus i=1-től M-ig
    Ha cs[Fut[i].a,Fut[i].b]>Fut[i].t
       akkor cs[Fut[i].a,Fut[i].b]:=Fut[i].t
  Ciklus vége
```

```
  Sorüres; PrSorba(h); tav[h]:=0
  Ciklus amíg nem üressor?
    Prsorból(i); min:=tav[i]
    Ciklus j=1-től n-ig
      Ha cs[i,j]<+∞ és tav[i]+cs[i,j]<tav[j]
        akkor PrSorba(j); tav[j]:=tav[i]+cs[i,j]
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

4. feladat: Ütemezés (15 pont)

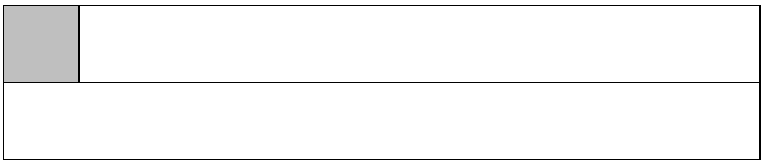
Úgy kell rendezni az alkatrészeket, hogy az első gépen rövid idejűek kerüljenek előre, a másodikon pedig a hosszú idejűek! Ekkor ugyanis az első gépen sok tartalék időnk lesz, amíg a második gép a lassú feladattal dolgozik. Azaz az i. alkatrész kerüljön sorra előbb, mint a j. alkatrész, ha `min(a[i],a[j],b[i],b[j])=a[i]` vagy `b[j]`. Az i. alkatrészig persze az első gép öszszeidejét legalább ki kell várni a második gépen az i. alkatrész feldolgozásának kezdetével.

Legyen `P[i]` a rendezés után az i. alkatrész eredeti sorszáma!

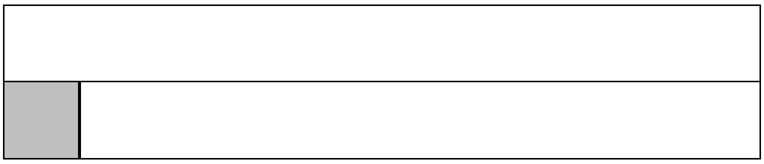
```
Ütemez(N,a,b,ered,P):
  Rendez
  at:=0; bt:=0
  Ciklus i=1-től N-ig
    at:=at+a[P[i]]; Ha bt<at akkor bt:=at
    bt:=bt+b[P[i]]
  Ciklus vége
  Ha at>bt akkor ered:=at különben ered:=bt
Eljárás vége.
```

5. feladat: Járdakövezés (15 pont)

A járdalefedést 1 vagy 2 méretű lappal indíthatjuk. Ha 1-essel indítjuk, akkor marad egy N hosszúságú olyan járda, amelynek az egyik bal sarka le van fedve:

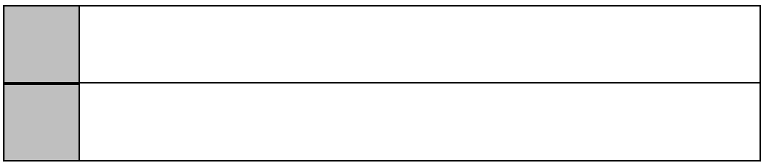


vagy ennek a tükörképe:

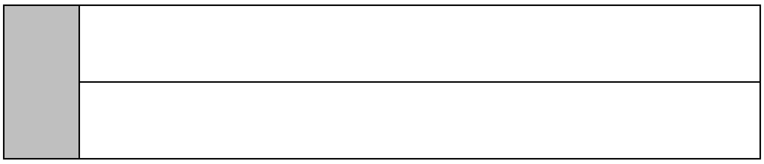


A kétféle kezdésű járdák között azonban van átfedés, ha az első két oszlopba 1-es méretű lapot teszünk.

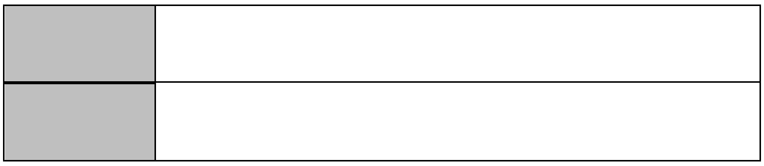
Ha két 1-essel indítunk, akkor N-1 hosszúságú járda marad:



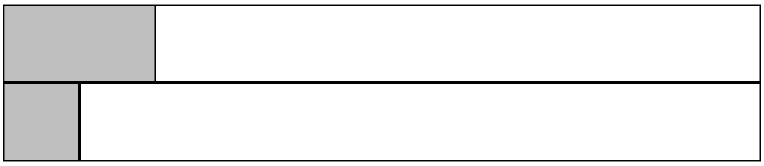
Ha függőlegesen 2-essel indítunk, akkor N-1 hosszúságú járda marad:



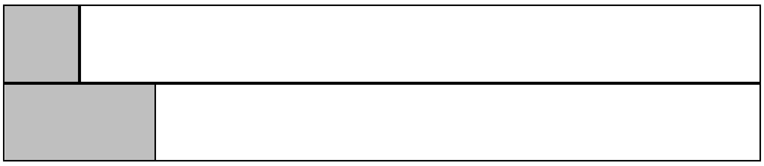
Ha vízszintesen két 2-essel indítunk, akkor N-2 hosszúságú járda marad:



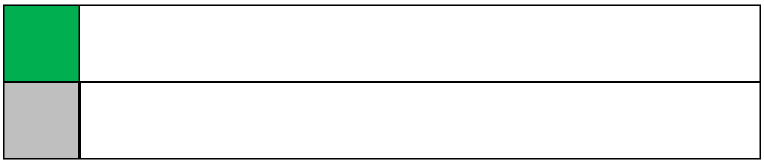
Ha egy 2-essel és egy 1-essel indítunk, akkor N-1 hosszúságú balsarok lefedett járda marad:



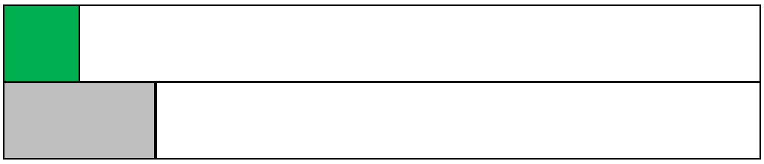
A tükörképe is lehet



A lefedett sarkú járdát 1-essel folytathatjuk:



vagy 2-essel folytathatjuk:



Azaz az N hosszúságú üres járda lefedései száma az N-1 hosszúságú üres járdák, valamint az N hosszúságú balsarok lefedett járdák lefedései számának összege. Az N hosszúságú balsarok lefedett járdák lefedései száma pedig az N-1 hosszúságú üres, illetve balsarok lefedett járdák lefedései száma.

```
Lefedésszám(N,Db):
  A[1]:=2; B[1]:=1
  A[2]:=7; B[2]:=3
  Ciklus i=3-tól N-ig
    A[i]:=A[i-2]+2*A[i-1]+2*B[i-1]
    B[i]:=A[i-1]+B[i-1]
  Ciklus vége
Eljárás vége.
```

2010. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Lövészverseny (25 pont)

Az első részfeladatban minden elemre meg kell számolni, hogy előtte hány nála nagyobb érték szerepelt!

A második részfeladatban maximumkiválasztás, a harmadikban minimumkiválasztás a feladat, de közben ki kell gyűjteni azokat, akik egyenlők a pillanatnyi maximummal; illetve minimummal.

A negyedik részfeladatban a maximálissal egyezőket kell kigyűjteni!

```
Lövész(n,pont,jobb,első,utolsó,győztes):
  jobb[1]:=0
  Ciklus i=2-től n-ig
    db:=0
    Ciklus j=1-től i-1-ig
      Ha pont[j]>pont[i] akkor db:=db+1
    Ciklus vége
  Ciklus vége
  max:=1; edb:=1; első[1]:=1
  Ciklus i=2-től n-ig
    Ha pont[i]≥pont[max] akkor edb:=edb+1; első[edb]:=i
    Ha pont[i]>pont[max] akkor max:=i
  Ciklus vége
  min:=1; udb:=1, utolsó[1]:=1
  Ciklus i=2-től n-ig
    Ha pont[i]≤pont[min] akkor udb:=udb+1; utolsó[udb]:=i
    Ha pont[i]<pont[min] akkor min:=i
  Ciklus vége
  gydb:=0;
  Ciklus i=1-től N-ig
    Ha pont[i]=pont[max] akkor gydb:=gydb+1; győztes[gydb]:=i
  Ciklus vége
eljárás vége.
```

2. feladat: Tördelés (25 pont)

Ha a szöveg rövidebb, mint a sorhossz, akkor kiírható. Ha az i. karaktertől az s-sel később levő szóköz, akkor addig kiírható, de a szóköz ne kerüljön be a következő sor elejére. Ha a sor végén lenne a szóköz, akkor is kiírható a sor. Egyéb esetben meg kell keresni az utolsó szó végét, ami még kifér a sorba!

```
Tördelés(n,szoveg,s,sdb,sor):
  i:=1; j:=s; sdb:=0
  Ciklus amíg i≤n
    Ha j≥n akkor
      sdb:=sdb+1
      Ciklus k=i-től n-ig
        sor[sdb]:=sor[sdb]+szoveg[k]
      Ciklus vége
      i:=n+1
```

```
    különben ha szoveg[j+1]=' ' akkor
      sdb:=sdb+1
      Ciklus k=i-től j-ig
        sor[sdb]:=sor[sdb]+szoveg[k]
      Ciklus vége
      i:=j+2; j:=i+s-1
    különben ha szoveg[j]=' ' akkor
      sdb:=sdb+1
      Ciklus k=i-től j-ig
        sor[sdb]:=sor[sdb]+szoveg[k]
      Ciklus vége
      i:=j+1; j:=i+s-1
    különben
      Ciklus amíg szoveg[j]≠' '
        j:=j-1
      Ciklus vége
      sdb:=sdb+1
      Ciklus k=i-től j-ig
        sor[sdb]:=sor[sdb]+szoveg[k]
      Ciklus vége
      i:=j+1; j:=i+s-1
    Elágazás vége
  Ciklus vége
Eljárás vége.
```

<u>3. feladat</u>: Verseny (25 pont)

A szélsőséges pontok kihagyásához maximum- és minimumkiválasztás kell. A legegyszerűbb lenullázni ezeket a pontokat. Figyelni kell arra, ha mindenki azonos pontot adott!

```
Kihagyás(n,m,pont,kihagy):
  Ciklus i=1-től N-ig
    Ha pont[i,1]<pont[i,2] akkor min:=1; max:=2
                          különben min:=2; max:=1
    Ciklus j=3-tól M-ig
      Ha pont[i,j]<pont[i,min] akkor min:=j
      Ha pont[i,j]>pont[i,max] akkor max:=j
    Ciklus vége
    kihagy[i,1]:=min; kihagy[i,2]:=max
    pont[i,min]:=0; pont[i,max]:=0
  Ciklus vége
Eljárás vége.
```

A pontszámok a lenullázott pontok miatt egyszerűen a mátrix sorösszegei.

```
Pontszámok(n,m,pont,psz):
  Ciklus i=1-től n-ig
    psz[i]:=0
    Ciklus j=1-től M-ig
      psz[i]:=psz[i]+pont[i,j]
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

A rendezéshez tárolni kell az egyes versenyzők sorszámait is, és a sorszámokat vinni kell rendezés közben! A helyezés megadásakor figyelni kell arra, hogy a versenyző pontszáma azonos-e az előzőével!

```
Helyezések(n,m,psz,hely):
  sorszám:=(1,…,n)
  Rendez(psz,sorszám)
  hely[1,1]:=1; hely[1,2]:=sorszám[1]; hely[1,3]:=psz[1]
  Ciklus i=2-től n-ig
    Ha psz[i]=psz[i-1] akkor hely[i,1]:=hely[i-1,1]
                      különben hely[i,1]:=i
    hely[i,2]:=sorszám[i]; hely[i,3]:=psz[i]
  Ciklus vége
Eljárás vége.
```

Kilencedik-tizedik osztályosok

<u>1. feladat</u>: Játék (19 pont)

Kapunk három intervallumhalmazt. Ezeket először logikai vektorokká alakítjuk, majd közös stratégiát alkotható játékospárok szerint megoldjuk.

```
Játék(adb,ha,bdb,hb,cdb,hc,ab,ac,bc):
  Átalakít(adb,ha,a); Átalakít(bdb,hb,b); Átalakít(cdb,hc,c)
  ab:=megold(a,b,c); ac:=megold(a,c,b); bc:=megold(b,c,a)
Eljárás vége.
```

```
Átalakít(db,h,a):
  a:=(hamis,…,hamis)
  Ciklus i=1-től db-ig
    Ciklus j=h[i].kezdet-től h[i].vég-ig
      a[j]:=igaz
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Egy jó intervallum ott kezdődik, ahol az első két játékos játszik, a harmadik pedig nem. Ott végződik, ahol az első nem játszik, vagy a második nem játszik, vagy a harmadik játszik.

```
megold(a,b,c,dbz,z):
  dbz:=0; van:=hamis
  Ciklus i=1-től n-ig
    Ha van akkor
      Ha nem a[i] vagy nem b[i] vagy c[i] akkor
          van:=hamis; z[dbz].veg:=i-1
    különben
      Ha a[i] és b[i] és nem c[i] akkor
          van:=igaz; dbz:=dbz+1; z[dbz].kezd:=i
  Ciklus vége
  Ha van akkor z[dbz].veg:=n
  megold:=z
Eljárás vége.
```

<u>2. feladat</u>: Térkép (16 pont)

A feladat egy módosított szélességi bejárás a P pontból a Q pontba. Két szomszédos pont között akkor van él a gráfban, ha a magasságuk különbsége legfeljebb H. Az él hossza pedig a magasságuk külüönbsége+1.

```
Bejárás(ps,po,qs,qo,n,m,t,h,mego):
  Prsorüres; tav[ps,po]:=0; Prsorba(ps,po)
  Ciklus amíg nem Üresprsor?
    Prsorból(i,j)
    Ha i>1 akkor Szomszéd(i,j,i-1,j)
    Ha j>1 akkor Szomszéd(i,j,i,j-1)
    Ha i<n akkor Szomszéd(i,j,i+1,j)
    Ha j<m akkor Szomszéd(i,j,i,j+1)
  Ciklus vége
Eljárás vége.
```

```
szomszed(i,j,k,l):
  a:=abs(t[i,j]-t[k,l])
  Ha a≤h akkor
    Ha tav[k,l]≥0 és tav[i,j]+1+a<tav[k,l]
       akkor tav[k,l]:=tav[i,j]+1+a; Prsorbanelőre(k,l)
    különben ha tav[k,l]<0
       akkor tav[k,l]:=tav[i,j]+1+a; Prsorba(k,l)
  Elágazások vége
Eljárás vége.
```

<u>3. feladat</u>: Vasúti kocsik rendezése (15 pont)

Az alábbi eljárást N-szer kell meghívni, az N bemeneti sorra.

Ha a sorozatban nincs 1-es, akkor a feladat megoldható (pl. 2-esek azonnal átmennek az A szakaszra, 3-asok az egyik, 4-esek a másik kitérőre. Ugyanez igaz a másik három számra is, sőt, ha bármelyik nincs az utolsó 1-es előtt, akkor is van megoldás.

Ha mind a 4 szám szerepel, akkor, ha az első 2-es az utolsó 4-es után van, a feladat akkor is megoldható (2-esek ugyanarra a kitérőre mennek, mint a 4-esek).

Ha az első 2-es és utolsó 4-es között van 3-as (az előzők miatt az első 2-es után valahol van 1-es), akkor nincs megoldás.

Ha az első 2-es előtt nincs 3-as (azaz az előzőek szerint az utolsó 4-es előtt sincsenek), akkor is van megoldás (a 4-esek után mennek a kitérőre), vagy az utolsó 4-es és 1-es között sincs 3-as, akkor is van megoldás.

```
Megoldás(m,log):
  ut1:=utolsó(1,1,m); el2:=első(2,1,ut1)
  ut4:=utolsó(4,1,ut1); el3:=első(3,1,ut1)
  Ha ut1=0 vagy el2=0 vagy el3=0 vagy ut4=0 akkor log:=igaz
  különben ha el2>ut4 akkor log:=igaz
  különben el3:=első(3,el2,ut4)
           Ha el3≠0 akkor log:=hamis
           különben ha első(3,1,el2)=0 vagy első(3,ut4.ut1)=0
                    akkor log:=igaz
           különben ha első(4,ut1,m)=0 vagy utolsó(2,ut1,m)=0
                    akkor log:=igaz
           különben log:=hamis
Eljárás vége.
```

Segédeljárásként meg kell keresnünk, az első, illetve utolsó valamilyen értéket a sorozatban!

```
első(mi,tól,ig):
  i:=tól
  Ciklus amíg i≤ig és mi≠x[i]
    i:=i+1
  Ciklus vége
  Ha i≤ig akkor első:=i különben első:=0
Függvény vége.
```

```
utolsó(mi,tól,ig):
  i:=ig
  Ciklus amíg i≥tól és mi≠x[i]
    i:=i+1
  Ciklus vége
  Ha i≥tól akkor utolsó:=i különben utolsó:=0
Függvény vége.
```

<u>4. feladat</u>: Gép kölcsönzés (25 pont)

A feladat dinamikus programozással oldható meg, `Opt(i,j)` legyen azon napok maximális száma, amire a gép kölcsönözhető az első `i` napon, az első `j` igényt figyelembe véve.

```
Megoldás(n,m,hat,nap):
  Opt[0..nap[1]-1,1]:=0
  Opt[nap[1]..m,1]:=nap[1]
  Ciklus i=2-től N-ig
    Opt[0,i]:=0
    Ciklus x=1-től M-ig
      Ha x<nap[i] akkor Opt[x,i]:=Opt[x,i-1]
      különben ha x≤hat[i] akkor
                  Opt[x,i]:=Opt[x,i-1]
                  Ha Opt[x,i]<Opt[x-nap[i],i-1]+nap[i]
                     akkor Opt[x,i]<Opt[x-nap[i],i-1]+nap[i]
      különben Opt[x,i]:=Opt[x-1,i]
    Ciklus vége
  Ciklus vége
  Megoldás:=Opt[M,N)
Eljárás vége.
```

Az `Opt` tömbből a megoldás lépései visszafejthetők.

```
Lépések(m,n):
  x:=M; i:=n; db:=0
  Ciklus amíg i>0 és x>0
    Ciklus amíg x>0 és Opt[x,i]=Opt[x-1,i]
      x:=x-1
    Ciklus vége
    Ha x>0 akkor Ciklus amíg i>0 és Opt[x,i]=Opt[x,i-1]
                   i:=i-1
                 Ciklus vége
                 Ha i>0 akkor db:=db+1
                              Telj[db].az:=i
                              Telj[db].kezd:=x-nap[i]+1
                              x:=x-nap[i]; i:=i-1
  Ciklus vége
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Játékos (18 pont)

Kapunk három intervallumhalmazt. Ezeket először logikai vektorokká alakítjuk, majd az egyedüli játékosok szerint megoldjuk.

```
Játék(adb,ha,bdb,hb,cdb,hc,ab,ac,bc):
  Átalakít(adb,ha,a); Átalakít(bdb,hb,b); Átalakít(cdb,hc,c)
  ab:=megold(a,b,c); ac:=megold(a,c,b); bc:=megold(b,c,a)
Eljárás vége.
```

```
Átalakít(db,h,a):
  a:=(hamis,…,hamis)
  Ciklus i=1-től db-ig
    Ciklus j=h[i].kezdet-től h[i].vég-ig
      a[j]:=igaz
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Egy jó intervallum ott kezdődik, ahol a `c` játékos játékos játszik, a másik kettő pedig nem. Ott végződik, ahol az első vagy a második játszik, vagy a harmadik nem játszik.

```
megold(a,b,c,dbz,z):
  dbz:=0; van:=hamis
  Ciklus i=1-től n-ig
    Ha van akkor
      Ha a[i] vagy b[i] vagy nem c[i]
         akkor van:=hamis; z[dbz].veg:=i-1
    különben
      Ha nem a[i] és nem b[i] és c[i]
         akkor van:=igaz; dbz:=dbz+1; z[dbz].kezd:=i
  Ciklus vége
  Ha van akkor z[dbz].veg:=n
  megold:=z
Eljárás vége.
```

2. feladat: Túra (15 pont)

A feladat megoldása azonos az előző korcsoport TÚRA feladatával, csak még a bejárt útvonalat is meg kell adni!

```
Bejárás(ps,po,qs,qo,n,m,t,h,mego):
  Prsorüres; tav[ps,po]:=0; PrSorba(ps,po); út[ps,po]:='.'
  Ciklus amíg nem Üresprsor?
    Prsorból(i,j)
    Ha i>1 akkor Szomszéd(i,j,i-1,j,'F')
    Ha j>1 akkor Szomszéd(i,j,i,j-1,'B')
    Ha i<n akkor Szomszéd(i,j,i+1,j,'L')
    Ha j<m akkor Szomszéd(i,j,i,j+1,'J')
  Ciklus vége
Eljárás vége.
```

```
Szomszéd(i,j,k,l,lépés):
  a:=abs(t[i,j]-t[k,l])
  Ha a≤h akkor
    Ha tav[k,l]≥0 és tav[i,j]+1+a<tav[k,l]
       akkor tav[k,l]:=tav[i,j]+1+a; út[k,l]:=lépés
             Prsorbanelőre(k,l)
    különben ha tav[k,l]<0
       akkor tav[k,l]:=tav[i,j]+1+a; út[k,l]:=lépés
             PrSorba(k,l)
  Elágazások vége
Eljárás vége.
```

```
Útkiírás(i,j):
  Ha út[i,j]≠'.' akkor
     Ha út[i,j]='F' akkor Útkiírás(i+1,j)
     különben ha út[i,j]='L' akkor Útkiírás(i-1,j)
     különben ha út[i,j]='B' akkor Útkiírás(i,j+1)
     különben Útkiírás(i,j-1)
     Ki: út[i,j])
Eljárás vége.
```

<u>3. feladat</u>: Vasúti kocsik rendezése (14 pont)

A feladat azonos az előző korcsoport vasúti kocsis feladatával.

<u>4. feladat</u>: Kötélhúzó verseny (14 pont)

Ez egy dinamikus programozásos feladat, a klasszikus „testvéries osztozkodás” feladat átszövegezése.

```
Megoldás(n,S):
  ms:=0
  Ciklus i=1-től n-ig
    ms:=ms+S[i]
  Ciklus vége
  fél:=ms div 2
  Ciklus x=0-tól fél-ig
    Ciklus k=1-től n/2-ig
      V[x,k,1]:=k=1 és x=S[1]
    Ciklus vége
  Ciklus vége
  Ciklus i=2-től n-ig
    Ciklus x=1-től fél-ig
      Ciklus k=2-től n/2-ig
        V[x,k,i]:=V[x,k,i-1] vagy S[i]≤x és V[x-S[i],k-1,i-1]
      Ciklus vége
    Ciklus vége
  Ciklus vége
  x:=fél
  Ciklus amíg nem V[x,n/2,n]
    x:=x-1
  Ciklus vége
  marad:=ms-2*x
```

```
  Cs:=(hamis,…,hamis)
  i:=N; k:=n/2
  Ciklus amíg x>0
    Ciklus amíg i>0 és V[x,k,i-1])
      i:=i-1
    Ciklus vége
    Cs[i]:=igaz; x:=x-S[i]; i:=i-1; k:=k-1
  Ciklus vége
Eljárás vége.
```

5. feladat: Legnagyobb négyzet (14 pont)

Könnyű lenne minden lehetséges pontnégyesre megnézni, hogy négyzetet alkotnak-e, de ennek futási ideje a pontok számának negyedik hatványával arányos, azaz nem kivárható.

Rendezzük a pontokat az x- és y-koordinátáik különbsége alapján az alábbi reláció szerint:

```
kisebb(p1,p2):
  kisebb:=p1.y-p1.x<p2.y-p2.x vagy
          p1.y-p1.x=p2.y-p2.x és p1.x<p2.x
Függvény vége.
```

Ezután minden pontpárhoz (ők lesznek a négyzet szemben levő sarkai) keresünk logaritmikus kereséssel másik két pontot, de csak olyanokra, amelyek x- és y-koordinátáik különbsége azonos (azaz lehetnek egy négyzet szemben levő sarkai).

```
Megoldás:
  Rendez; maxi:=0; q1.az:=0
  Ciklus i=2-től N-2-ig
    ii:=i
    Ciklus amíg ii<N és P[ii].y-P[ii].x=P[ii+1].y-P[ii+1].x
      ii:=ii+1
    Ciklus vége
    Ha i≠ii akkor
       Ciklus j=i-től ii-1-ig
         Ciklus jj=j+1-től ii-ig
           p1:=P[j]; p3:=P[jj]
           p2.x:=p2.x; p2.y:=p1.y
           p2.az:=Keres(1,i-1,p2)
           Ha p2.az≠0 akkor p4.x:=p1.x; p4.y:=p3.y
                            p4.az:=Keres(i+1,N,p4)
                            Ha p4.az≠0 akkor
                               Ha (p2.x-p1.x)²>maxi akkor
                                  maxi:=(p2.x-p1.x)²
                                  q1:=p1;q2:=p2;q3:=p3;q4:=p4
         Ciklus vége
       Ciklus vége
  Ciklus vége
Eljárás vége.
```

2011. Első forduló

Ötödik-nyolcadik osztályosok

Számítógép nélküli feladatok

1. feladat: Hibakeresés (25 pont)

`Ciklus i=1-től` **`N`**`-ig`	2+3 pont
`Ciklus amíg j≤M` **`és`**	2+3 pont
`D(i,j)>0`	2+3 pont
`Ha` **`j`**	2+3 pont
`>``M akkor Ki: i`	2+3 pont

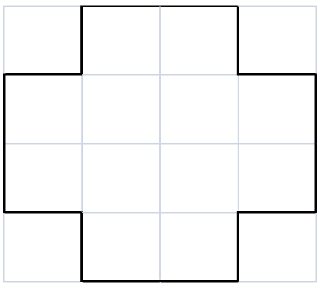
2. feladat: Mit csinál (20 pont)

A. H=1	3 pont
B. H=3	3 pont
C. H=4	3 pont
D. H=2	3 pont

E. A maximális értékű elem sorszámát kapjuk H-ban; ha több egyforma is van, akkor az elsőt
4+4 pont

3. feladat: Robot (25 pont)

Megjegyzés: a sarkokon a robot a valóságban ívben fordul, azaz lekerekített sarkú négyzetek a megoldások, de a mellékelt, szögletese ábrát is elfogadjuk.



A.	4 szimmetrikus elemből áll; visszatér a kezdő helyre	3+3 pont
	minden elem 3 egyenesből áll; jó szögben fordulva	3+3 pont
	az egyenesek 5 és 10 egység méretűek (azaz a rövidebb hossza kb. fele a hosszabb hosszának)	3 pont
B.	Legalább 1 négyzetet bejár	3 pont
	5 négyzetet jár be	3 pont
	a négyzetek elhelyezése jó	4 pont

Számítógépes feladat – VÁLASZTHATÓ

4. feladat: Balaton (30 pont)

Három programozási tételt kell alkalmazni:

- nem 0 értékek megszámolása;
- egy nem 0 érték megkeresése;
- legnagyobb érték meghatározás (ha nem nulla).

```
Fagyott napok(n,jég,db):
  db:=0
  Ciklus i=1-től n-ig
    Ha jég[i]>0 akkor db:=db+1
  Ciklus vége
Eljárás vége.
```

```
Korcsolyázás(n,jég,van,nap):
  i:=1
  Ciklus amíg i≤n és jég[i]=0
    i:=i+1
  Ciklus vége
  van:=i≤n
  Ha van akkor nap:=i
Eljárás vége.
```

```
Legvastagabb(n,jég,van,nap):
  nap:=1
  Ciklus i=2-től n-ig
    Ha jég[i]>jég[nap] akkor nap:=i
  Ciklus vége
  van:=jég[nap]>0
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Számok (18 pont)

```
Eljárás(N,M,D,db,mad):
  db:=0
  Ciklus j=1-től M-ig                                   2 pont
    i:=1
    Ciklus amíg i≤N és nem(van(i) és D(i,j)=0)       2+2+2 pont
      i:=i+1
    Ciklus vége
    Ha i>N akkor db:=db+1; mad(db):=j                   2+2 pont
  Ciklus vége.
Eljárás vége.
```

```
van(i):
  k:=1
  Ciklus amíg k≤M és D(i,k)=0                           2+2 pont
    k:=k+1
  Ciklus vége
  van:=(k≤M)                                            2 pont
Függvény vége.
```

2. feladat: Mit csinál (20 pont)

A. C=16 — 3 pont

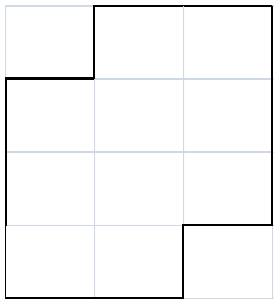
B. C=80 — 3 pont

C. C=150 — 3 pont

D. C=A*B — 5 pont

E. A*B=C+D*E — 6 pont
(A D és az E részfeladatra ezzel egyenértékű szöveges megoldások is elfogadhatók.)

3. feladat: Robot (25 pont)



A. 2-2 szimmetrikus elemből áll; visszatér a kezdő helyre 3+3 pont

az elemek jó darabszámú egyenesből állnak; jó szögben fordulva 3+3 pont

az egyenesek 5 és 15 egység méretűek (azaz a rövidebb hossza kb. harmada a hosszabb hosszának) 3 pont

B. A belső négyzetet bejárja 3 pont

a 4 külső négyzet lyukas 3 pont

a négyzetek elhelyezése jó 4 pont

4. feladat: Adatok (28 pont)

A.	(0,0,1,0,0,0)	1 pont
	(0,0,1,2,0,0)	1 pont
	(0,3,1,2,0,0)	2 pont
	(0,3,1,2,3,0)	2 pont
	(6,3,1,2,3,0)	2 pont
B.	(0,0,1,0,0,0)	1 pont
	(0,0,1,2,0,0)	1 pont
	(0,1,1,2,0,0)	2 pont
	(0,1,1,2,1,0)	2 pont
	(6,1,1,2,1,0)	2 pont
C.	0; 1; 2; 6; 10; 15	1+1+2+2+2+2 pont

5. feladat: Játék (21 pont)

A. 4	3 pont
B. 1024	4 pont
C. 49	4 pont
D. 823543	5 pont
E. 19683	5 pont

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Számok (14 pont)

```
Eljárás(N,M,D,db,mad):
  db:=0
  Ciklus j=1-től M-ig
    i:=1
    Ha j=i akkor i:=i+1
    Ciklus amíg i≤M és nem jó(i,j)                     3 pont
      i:=i+1
      Ha j=i akkor i:=i+1
    Ciklus vége
    Ha i≤M akkor db:=db+1; mad(db):=j                  2 pont
  Ciklus vége
Eljárás vége.
```

```
jó(i,j):                                                  3+3+3 pont
  k:=1
  Ciklus amíg k≤N és (D(k,i)>0 és D(k,j)>0 vagy D(k,j)=0)
    k:=k+1
  Ciklus vége
  jó:=(k>N)
Függvény vége.
```

2. feladat: Mit csinál (17 pont)

A. Y=4 — 2 pont

B. Y=2 — 2 pont

C. Y=5 — 2 pont

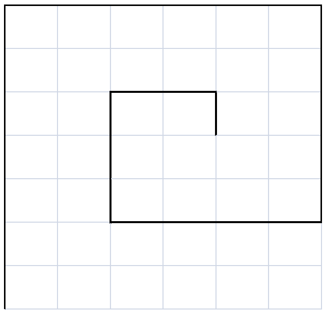
D: Y az X négyzetgyökének egész része — 5 pont

E. A egyesével nő (0,1,2,...), B páratlanok (1,3,5, ...), C négyzetszámok (0,1,4,9,...) — 2+2+2 pont

3. feladat: Robot (22 pont)

A. spirál 7 elemből — 2 pont
a párhuzamos oldalak távolsága állandó; jó szögben fordulva — 3+2 pont
az egyenesek 10 egység méretűek; a hosszuk 10 egységgel nö-vekszik — 3+2 pont

B. Legalább 1 négyzetet bejár — 3 pont
5 négyzetet jár be — 3 pont
a négyzetek elhelyezése jó — 4 pont



4. feladat: Adatok (26 pont)

A. (0,0,1,0,0,0) — 1 pont
(0,0,1,2,0,0) — 1 pont
(0,1,1,2,0,0) — 2 pont
(0,1,1,2,1,0) — 2 pont
(6,1,1,2,1,0) — 2 pont

B. 0; 1; 2; 6; 10; 15 — 1+1+1+1+1+1 pont

C. (0,0,1,0,0,0) — 1 pont
(0,0,1,2,0,0) — 1 pont
(0,1,1,2,0,0) — 2 pont
(0,1,1,1,0,0) — 2 pont
(5,1,1,1,0,0) — 2 pont
(5,5,5,1,0,0) — 2 pont
(5,5,5,1,6,0) — 2 pont

5. feladat: Fűrészmalom (21 pont)

A. 13 — 5 pont

B. 7, 13 — 5 pont

C. 6, 8, 13 — 5 pont

D. 6, 8, 13, 14 — 6 pont

2011. Második forduló

Ötödik-nyolcadik osztályosok

1. feladat: Sípálya (25 pont)

Három programozási tételt kell alkalmazni:

- maximumkiválasztás;
- 100-nál nagyobb érték keresése;
- a 0 értékűek kiválogatása

```
Legvastagabb hó(n,v,max):
  max:=1
  Ciklus i=2-től n-ig
    Ha v[i]>v[max] akkor max:=i
  Ciklus vége
Eljárás vége.
```

```
Sípálya(n,v,van,p):
  p:=1
  Ciklus amíg p≤n éa v[p]<100
    p:=p+1
  Ciklus vége
  van:=p≤n
Eljárás vége.
```

```
Sípályák(n,v,db,w):
  db:=0
  Ciklus i=1-től n-ig
    Ha v[i]=0 akkor db:=db+1; w[db]:=i
  Ciklus vége
Eljárás vége.
```

2. feladat: Pénz (30 pont)

Amilyen pénzérméink vannak, azok az összegek fizethetők ki egy pénzérmével. A két pénzérmések azok, amelyek eggyel kifizethetők, megnövelve bármely pénzérmével. A hárommal kifizethetők hasonlóan számíthatók.

```
Pénz(m,n,p):
  lehet:=(hamis,...,hamis)
  Ciklus i=1-től n-ig
    lehet[p[i]]:=igaz
  Ciklus vége
  kiir(1)
  Ciklus j=m-től 1-ig -1-esével
    Ciklus i=1-től n-ig
      Ha lehet[j] és p[i]+j≤m akkor lehet[p[i]+j]:=igaz
    Ciklus vége
  Ciklus vége
  kiir(2)
```

```
  Ciklus j=m-től 1-ig -1-esével
    Ciklus i=1-től n-ig
      Ha lehet[j] és p[i]+j≤m akkor lehet[p[i]+j]:=igaz
    Ciklus vége
  Ciklus vége
  kiir(3)
Eljárás vége.
```

3. feladat: Nyelv (20 pont)

Az IF szónál elágazás kezdődik, a FI szónál elágazás végződik – ezek alapján számolhatjuk a nyitott elágazások számát. Hibalehetőségek:

- menet közben a megkezdett elágazások száma negatívvá válik – FI jött IF nélkül;
- a végén a megkezdett elágazások száma nem 0 – IF-ből több van, mint FI-ből.

Ha a végén a megkezdett elágazások száma 0, akkor helyes a program.

```
Nyelv(n,szo,db):
  db:=0; i:=1
  Ciklus amíg i≤n és db≥0
    Ha szo[i]='IF' akkor db:=db+1
    Ha szo[i]='FI' akkor db:=db-1
    i:=i+1
  Ciklus vége
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Falu (18 pont)

A gráfot csúcsmátrix-szal ábrázoljuk, tárolva minden pont fokszámát is (`g`,`fok`). Az első feladat az 1 fokszámúak száma, a második a maximális fokszámú pont, a harmadik pedig a nem szomszédos pontpárok közötti egyetlen közbülső pontokra legkisebb távolságú pár megadása.

```
Első(n,fok,db):
  db:=0
  Ciklus i=1-től n-ig
    Ha fok[i]=1 akkor db:=db+1
  Ciklus vége
Eljárás vége

Második(n,fok,max):
  max:=1
  Ciklus i=2-től n-ig
    Ha fok[i]>fok[max] akkor max:=i
  Ciklus vég
Eljárás vége.
```

```
Harmadik(n,g,c,d):
  min:=+∞
  Ciklus i=1-től n-1-ig
    Ciklus j:=i+1-től n-ig
      Ha i≠j és g[i,j]=+∞ akkor
         Ciklus k=1-től n-ig
           Ha g[i,k]<+∞ és g[k,j]<+∞ akkor
              Ha g[i,k]+g[k,j]<min akkor min:=g[i,k]+g[k,j]
                                          c:=i; d:=j
         Ciklus vége
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

2. feladat: Fa (18 pont)

Beolvasáskor tegyünk `s[0]`-ba (-jelet, `s[n+1]`-be pedig) -jelet!

Az első részfeladat a fa magassága. Ez egyszerűen a zárójelek legmélyebb egymásbaágyazása.

```
Magasság(s,index):
  max:=0; a:=0
  Ciklus i=0-tól n-ig
    Ha s[i]='(' akkor a:=a+1
    különben ha s[i]=')' akkor a:=a-1
    Ha a>max akkor max:=a
  Ciklus vége
  Magasság:=max
Függvény vége.
```

A második részfeladat a nyitózárójelek száma:

```
Elágazásszám(s):
  db:=0
  Ciklus i=1-től n-ig
    Ha s[i]='(' akkor db:=db+1
  Ciklus vége
  Elágazásszám:=db
Függvény vége.
```

A harmadik részfeladat az adott szinten belül az azonos szinten levő nyitózárójelek száma:

```
Maxelágazás(s):
  max:=0; db:=0
  Ciklus i=1-től n-ig
    Ha s[i]='(' akkor db:=db+1; t[db]:=0; t[db-1]:=t[db-1]+1
                      Ha t[db-1]>max akkor max:=t[db-1]
    különben ha s[i]=')' akkor db:=db-1
  Ciklus vége
  Maxelágazás:=max
Függvény vége.
```

3. feladat: Pakolás (18 pont)

Haladjunk balról jobbra, amíg a láda méret növekszik! Ezek biztosan rátehetők arra, ameddig elértünk, de a tőle jobbra csökkenő sorrendben levők is (hacsak nincs két egyforma a két oldalon). A maradékra ugyanezt az eljárást folytassuk!

```
Pakol(m):
  m:=0; i:=1; a[n+1]:=0
  Ciklus amíg i≤n
    bal:=i
    Ciklus amíg i<n és a[i]<a[i+1]
      i:=i+1
    Ciklus vége
    balra:=i-1; j:=i+1; t:=a[i]
    Ciklus amíg bal≤balra
      Ha a[j]<t és a[balra]<a[j] akkor t:=a[j]; j:=j+1
      különben t:=a[balra]; balra:=balra-1
    Ciklus vége
    Ciklus amíg j≤n és t>a[j]
      t:=a[j]; j:=j+1
    Ciklus vége
    m:=m+1; i:=j
  Ciklus vége
Eljárás vége.
```

4. feladat: Játék (21 pont)

Legyen `L[x,y]` az (x,y) pozícióról az (n,m) pozícióra eljutások maximális száma! A csapda helyről ez 0, minden más helyről az alsó és a jobboldali szomszédjaiból az eljutások maximális számának összege.

```
Játék(n,m,L,Eredmény):
  L[1..n,m+1]:=(0,...,0); L[n,m]:=1
  Ciklus y=m-1-től 1-ig -1-esével
    Ha L[n,y]=1 akkor L[n,y]:=0
             különben L[n,y]:=L[n,y+1]
  Ciklus vége
  Ciklus x=n-1-től 1-ig -1-esével
    Ciklus y=m-től 1-ig -1-esével
      Ha L[x,y]=1 akkor L[x,y]:=0
               különben L[x,y]:=L[x+1,y]+L[x,y+1]
    Ciklus vége
  Ciklus vége
  eredmény:=L[1,1]
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Falvak (15 pont)

A `t` tömb tartalmazza a gráfot, ahol `t[i,j]` az `i`. település `j`. szomszédjának sorszáma! A harmadik részfeladathoz `táv[i,j]` legyen ennek a településnek az `i`.településtől vett távolsága!

Az első részfeladatnál az 1 fokszámú pontokból megyünk addig, amíg 2 fokszámú pontokon megyünk keresztül, majd közülük kiválasztjuk a leghosszabb utat.

```
Zsákút(n,fok,t,max):
  max:=0
  Ciklus i=1-től n-ig
    Ha fok[i]=1
       akkor a:=t[i,1].tav; j:=t[i,1].ind; e:=i
             Ciklus amíg fok[j]=2
               Ha e=t[j,1].ind akkor
                        a:=aa+t[j,2].tav; e:=j; j:=t[j,2].ind
               különben a:=aa+t[j,1].tav; e:=j; j:=t[j,1].ind
             Ciklus vége
             Ha a>max akkor max:=a
  Ciklus vége
Eljárás vége.
```

A második részfeladat az összes maximális fokszámú pont megadásáról szól.

```
Maximálisak(n,fok,db,b):
  db:=1; b[1]:=1
  Ciklus i=2-től n-ig
    Ha fok[i]>fok[t[db]] akkor db:=1; t[1]:=i
    különben ha fok[i]=fok[t[dbb]] akkor db:=db+1; t[b]:=i
  Ciklus vége
Eljárás vége.
```

A harmadik részfeladatban minden pontra számoljuk ki a hozzá legközelebb levő pontot, majd ezek maximumát kell venni!

```
Távoli(n,fok,t,max):
  Ciklus i=1-től N-ig
    l[i]:=t[i,1]
    Ciklus j=2-től fok[i]-ig
      Ha táv[i,j]<l[i] akkor l[i]:=táv[i,j]
    Ciklus vége
  Ciklus vége
  max:=1
  Ciklus i=2-től n-ig
    Ha l[i]>l[max] akkor max:=i
  Ciklus vége
Eljárás vége.
```

2. feladat: Fa (15 pont)

Beolvasáskor tegyünk `s[0]`-ba `(`-jelet, `s[n+1]`-be pedig `)`-jelet!

Az első részfeladat a fa magassága. Ehhez számolni kell az egyes szakaszokon levő X betűk számát, az elágazásokon belül pedig a legmagasabb ág magasságát!

```
Magasság(s,index):
  ah:=0; am:=0
  Ha s[index]=')' akkor index:=index+1
  különben Ciklus amíg s[index]='X'
             ah:=ah+1; index:=index+1
           Ciklus vége
           am:=0
           Ciklus amíg s[index]≠')'
             index:=index+1; a:=Magasság(s,index)
             Ha a>am akkor am:=a
           Ciklus vége
           index:=index+1
  Magasság:=ah+am
Függvény vége.
```

A második részfeladat a nyitózárójelek száma:

```
Elágazásszám(s):
  db:=0
  Ciklus i=1-től n-ig
    Ha s[i]='(' akkor db:=db+1
  Ciklus vége
  Elágazásszám:=db
Függvény vége.
```

A harmadik részfeladat az egymásután levő X betűk száma:

```
Elágazásnélküli(s):
  max:=0; db:=0
  Ciklus i=0-tól n-ig
    Ha s[i]∈['(',')'] akkor Ha db>max akkor max:=db
    Ha s[i]='(' akkor db:=0
    különben ha s[i]='X' akkor db:=db+1
  Ciklus vége
  Ha db>max akkor max:=db
  Elágazásnélküli:=max
Függvény vége.
```

3. feladat: Takar (15 pont)

Sorba kellene rendezni az egyes épületeket irányszög szerint. Ehelyett azonban egyszerűbb és gyorsabb minden kezdőszöghöz (0 és 90 közötti egész szám) tárolni az ott kezdődő épület szögtartományának végét. Ha több is kezdődne ugyanott, akkor a nagyobbat.

```
Takart(vége,m):
  m:=0; első:=0; utolsó:=0
  Ciklus i=0-tól 90-ig
    Ha vége[i]>0 akkor
       Ha i≤utolsó akkor
          Ha vége[i]>utolsó akkor utolsó:=vége[i]
       különben
         m:=m+utolsó-első; első:=i; utolsó:=vége[i]
  Ciklus vége
  m:=m+utolsó-első
Eljárás vége.
```

4. feladat: Pakolás (15 pont)

A feladat azonos az előző korcsoport pakolás feladatával.

5. feladat: Játék (15 pont)

Számítsuk ki minden lehetséges mezőre, hogy addig eljutva mennyi a maximálisan begyűjthető kincsek száma, ha lentről, illetve, ha fentről léptünk be! Ha a csapda mezőkhöz nagy negatív számot rendelünk

```
Kincsek:
  L(N+1,2..M):=-1; L(2..N,0):=-1; L(N+1,1):=0; L(1,0):=0
  F(0,2..M):=-1; F(2..N,0):=-1; F(0,1):=0; F(1,0):=0
  Ciklus j=1-től M-ig
    Ciklus i=1-től N-ig
      Ha Kincs(i,j)=-1 akkor F[i,j]:=-N*M
      különben
            F(i,j):=max(F(i-1,j),L(i,j-1),F(i,j-1))+Kincs[i,j]
    Ciklus vége
    Ciklus i=N-től 1-ig -1-esével
      Ha Kincs(i,j)=-1 akkor L[i,j]:=-N*M
      különben
            L(i,j):=max(L(i+1,j),L(i,j-1),F(i,j-1))+Kincs[i,j]
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Az útkiírás a jobb alsó sarokból visszafelé történhet (ha nem negatív):

```
ÚtkiírásF(i,j,irány):
  Ha i≠1 vagy j≠1 akkor
     Ha F[i,j]=F[i-1,j] akkor ÚtkiírásF(i-1,j); Ki: ’L’
     különben ha F[i,j]=F[i,j-1] akkor ÚtkiírásF(i,j-1); Ki: ’J’
     különben ÚtkiírásL(i,j-1); Ki: ’J’
Eljárás vége.
```

```
ÚtkiírásL(i,j,irány):
  Ha i≠1 vagy j≠1 akkor
     Ha L[i,j]=L[i+1,j] akkor ÚtkiírásL(i+1,j); Ki: ’F’
     különben ha L[i,j]=L[i,j-1] akkor ÚtkiírásL(i,j-1); Ki: ’J’
     különben ÚtkiírásF(i,j-1); Ki: ’J’
Eljárás vége.
```

2011. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Időjárás (25 pont)

Az első részfeladatban egy K hosszúságú csupa negatív számot tartalmazó intervallumot kell megadni.

```
Folyamatos(n,max,van,kezdet,vég):
  kezdet:=0; vég:=0; i:=1
  Ciklus amíg i≤n és vég=0
    Ha max[i]≥0 akkor kezdet:=0
    különben ha kezdet=0 akkor kezdet:=i
    különben ha kezdet>0 és i-kezdet=k-1 akkor vég:=i
    i:=i+1
  Ciklus vége
  van:=(kezdet>0)
Eljárás vége.
```

Egy egyszerű maximumkiválasztás lehetne, csak a feltétele bonyolult, két szomszédos nap maximuma és minimuma különbségét kell figyelni!

```
Legnagyobb változás(n,max,min,nap):
  v:=-1
  Ciklus i=2-től n-ig
    Ha max[i]-min[i-1]>v akkor nap:=i-1; v:=max[i]-min[i-1]
    különben ha max[i-1]-minh[i]>v
                     akkor nap:=i-1; v:=max[i-1]-min[i]
  Ciklus vége
Eljárás vége.
```

A harmadik feladat egy összegzés, majd egy maximumkiválasztás.

```
Melegek(n,max,min,db,t):
  s:=0
  Ciklus i=1-től n-ig
    s:=s+min[i]+max[i]
  Ciklus vége
  db:=0
  Ciklus i=1-től n-ig
    Ha min[i]*2*n>s akkor db:=db+1; t[db]:=i
  Ciklus vége
Eljárás vége.
```

2. feladat: Webcím (25 pont)

Jó tesztek

http://valami.hu	1 pont
https://valami.hu/valami	1 pont
valami.hu	1 pont
http://valami.hu:80/valami	2 pont
https://valami.hu/valami?a=b	2 pont
http://valami.hu/valami?a=&c=b	2 pont

Hibás tesztek

http után nincs : (http//valami.hu/)	2 pont

http: után nincs // (http:valami.hu)	2 pont
nincs http, de van : vagy / (://valami.hu)	2 pont
csak egy / van a http: után (https:/valami.hu)	2 pont
két / között nincs semmi (http://valami.hu//	2 pont
hibás karakter (http:valami.hu\valami)	2 pont
a : nem a célgép tartományneve mögött van (https://valami.hu/valami:80)	2 pont
a : után nem port azonosító szám jön (http://valami.hu:valami)	2 pont

3. feladat: Elszigetelt falu (25 pont)

Minden falura számoljuk ki a legközelebbi szomszéd távolságát, majd ezek közül vegyük a legnagyobbat! Ehhez az utak beolvasása után a gráfot nem kell felépíteni, az utak alapján az eredmény kiszámolható.

```
Elszigetelt(n,m,kezdet,vég,e):
  lk:=(0,…,0)
  Ciklus i=1-től m-ig
    Ha lk[kezdet[i]]>hossz[i] akkor lk[kezdet[i]]:=hossz[i]
    Ha lk[vég[i]]>hossz[i] akkor lk[vég[i]]:=hossz[i]
  Ciklus vége
  e:=1
  Ciklus i=2-től n-ig
    Ha lk[i]>lk[e] akkor e:=i
  Ciklus vége
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Fénykép (19 pont)

Akkor fényképezzünk, amikor feltétlenül szükséges! Ez azt jelenti, hogy amikor elmenne az első ember, aki még nem volt rajta egy fényképen sem, akkor kezdődik egy fényképezési intervallum.

Az első változatban rendezzük az intervallumokat:

```
Kiválogatás(N,E,T,K,Db,X):
  Rendezés(N,E,T,S)
  Db:=1; X(Db):=S(1); j:=1
  Ciklus i=2-től N-ig
    Ha E(i)≥T(j)+K akkor Db:=Db+1; X(Db):=S(i); j:=i
  Ciklus vége
Eljárás vége.
```

A második változatban létrehozunk egy tömböt (Max elemszámú), amelynek távozási idővel indexelt elemeiben az érkezési idők vannak.

```
Kiválogatás(N,E,T,K,Db,X):
  Kezd:=(0,…,0)
  Ciklus i=1-től N-ig
    Ha Kezd[T[i]]<E[i] akkor Kezd[T[i]]:=E[i]
  Ciklus vége
  Db:=0; vég:=0
  Ciklus i=1-től Max-ig
    Ha Kezd[i]>0 és Kezd[i]≥vég) akkor db:=db+1; X[db]:=i
                                       vég:=i+K
  Ciklus vége
Eljárás vége.
```

2. feladat: Sudoku (15 pont)

A feladat visszalépéses kereséssel oldható meg! Az M vektorba töltsük be a bemeneti értékeket, sorfolytonosan! Tárolni kell minden értékre, hogy az adott sorban, oszlopban, illetve sarokban előfordulhat-e még (`Sor`, `Oszlop`, `Sarok` logikai tömbök, kezdetben csupa igaz értékekkel)!

A sorfolytonos elhelyezésben ezek a sarkok sorszámai: (1,1,2,2,1,1,2,2,3,3,4,4,3,3,4,4).

Ha a kiíró eljárásból visszatérünk, akkor az összes megoldást is megkapjuk.

```
Keres(i):
  Ha M[i]>0 akkor
     Ha Lehet(i,M[i]) akkor X[i]:=M[i]
                            Ha i=16 akkor Kiír
                            Keres(i+1); Vissza(i,M[i])
     különben Ciklus j=1-től 4-ig
                Ha Lehet(i,j) akkor X[i]:=j
                                    Ha i=16 akkor Kiír
                                    Keres(i+1); Vissza(i,j)
              Ciklus vége
Eljárás vége.
```

```
Lehet(i,j):
  si:=(i-1) div 4 +1; oi:=i mod 4; Ha oi=0 akkor oi:=4
  ki:=sorszám[i]
  Ha Sor[si,j] és Oszlop[oi,j] és Sarok[ki,j] akkor
     Sor[si,j]:=hamis; Oszlop[oi,j]:=hamis; Sarok[ki,j]:=hamis
     Lehet:=igaz
  különben Lehet:=hamis
Függvény vége.
```

```
Vissza(i,j):
  si:=(i-1) div 4 +1; oi:=i mod 4; Ha oi=0 akkor oi:=4
  ki:=sorszám[i]
  Sor[si,j]:=igaz; Oszlop[oi,j]:=igaz; Sarok[ki,j]:=igaz
Eljárás vége.
```

3. feladat: Jelentés (18 pont)

Tároljuk mindenkiről, hogy ki a főnöke (`F[i]`)!

A távolság rekurzív összefüggést memorizálással oldjuk meg, hogy mindenkire csak egyszer számoljunk távolságot! A `t` tömb összes eleme legyen negatív (még nem számoltuk ki őket), az 1. eleme pedig 0 (a főnök távolsága magától 0)!

```
táv(x):
  Ha t[x]<0 akkor t[x]:=táv(F[x])+1
  táv:=t[x]
Függvény vége.
```

```
leghosszabb(n,F,max,maxi):
  F[1]:=0; t[1]:=0; max:=0; t[2..n]:=-1
  Ciklus i=2-től n-ig
    x:=táv(i)
    Ha x>max akkor max:=x; maxi:=i
  Ciklus vége
Eljárás vége.
```

4. feladat: Csapat (20 pont)

A megoldás egy gráf szélességi bejárás, ahol az egyik csapatbelieket pirosra, a másikbelieket kékre színezzük. A feladat nem megoldható, ha valakinek vele egyező színű szomszédja keletkezne.

```
Csapatok:
  Szín:=(fehér,…,fehér); Jó:=igaz
  Ciklus x=1-től n-ig
    Ha Szín[x]=fehér akkor
       Szín[x]:=piros; Sorüres; Sorba(x)
       Ciklus amíg nem üressor? és Jó
         p:=Sorból
         Ciklus i=1-től Fok[p]-ig
           Ha Szín[G[p,i]]=fehér akkor
              Ha Szín[p]=piros akkor Szín[G[p,i]]:=kék
                            különben Szín[G[p,i]]:=piros
              Sorba(G[p,i])
           különben ha Szin[G[p,i]]=Szin[p] akkor Jó:=hamis
         Ciklus vége
       Ciklus vége
  Ciklus vége
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Párok (15 pont)

Rendezzük sorba távozási idő szerint az adatainkat! Ha valakinek még nincs párja a fényképezkedéshez, akkor válasszunk olyat párjának, aki ott van és a leghamarabb menne el!

```
Kiválogatás(N,E,T,Db,Pár):
  S:=(1,…,N); Rendezés(N,E,T,S); Db:=0
  Ciklus i=2-től N-ig
    j:=1
    Ciklus amíg j<i és nem(Pár(j)=0 és E(i)<T(j))
      j:=j+1
    Ciklus vége
    Ha j<i akkor Pár(j):=i; Pár(i):=j; Db:=Db+1
  Ciklus vége
Eljárás vége.
```

2. feladat: Sudoku (15 pont)

A feladat majdnem azonos az előző korcsoport feladatával. Annyi a különbség, hogy ha egy megoldást megtalálunk, akkor nem kiírni kell, hanem egy számlálót növelni!

```
Keres(i):
  Ha M[i]>0 akkor
     Ha Lehet(i,M[i])
        akkor X[i]:=M[i]
              Ha i=16 akkor Db:=Db+1 különben Keres(i+1)
              Vissza(i,M[i])
     különben Ciklus j=1-től 4-ig
                Ha Lehet(i,j)
                   akkor X[i]:=j
                         Ha i=16 akkor Db:=Db+1
                              különben Keres(i+1)
                         Vissza(i,j)
              Ciklus vége
Eljárás vége.
```

3. feladat: Kemence (15 pont)

Számítsuk ki az első `i` tárgy kiégetéséhez szükséges időt (`Idő(0)=0`)! Figyelembe kell venni mindegyik vele együtt égetendő minimális és maximális égetési idejét, valamint a határidejét is!

```
Kemence:
  Idő[0]:=0
  Ciklus i=1-től n-ig
    min:=Korsó[i].min; max:=Korsó[i].max; hidő:=Korsó[i].hidő
    Idő[i]:=Idő[i-1]+Korsó[i].min+1; Ut[i]:=0
    Ut[i]:=i; ii:=i-1
    Ciklus amíg ii>0 és i-K<ii
      Ha min<Korsó[ii].min akkor min:=Korsó[ii].min
      Ha Korsó[ii].max<max akkor max:=Korsó[ii].max
      Ha Korsó[ii].hidő<hidő akkor hidő:=Korsó[ii].hidő
      Ha min≤max és Idő[ii-1]+min+1≤hidő és
         Idő[ii-1]+min+1<Idő[i]
         akkor Idő[i]:=Idő[ii-1]+min+1; Ut[i]:=ii
      ii:=ii-1
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

```
Kiírás:
  i:=n; m:=0
  Ciklus amíg i>0
    m:=m+1; ii:=Ut[i]; mego[m].tol:=ii; mego[m].ig:=i
    i:=ii-1
  Ciklus vége
Eljárás vége.
```

4. feladat: Képzés (15 pont)

Dolgozóból biztosan kevesebb van, mint főnökükből, így ne menjen továbbképzésre az, akinek nincs beosztottja! A megoldás modellje egy nembináris fa. Színezzük szürkére azokat, akiknek nem kell képzésre menniük, feketére azokat, akiknek kell! A fa levelei szürkék lesznek. A fát alulról felfelé járjuk be, megállunk, ha olyan ponthoz értünk, ahol már jártunk.

```
Képzés:
  M:=1; Szín[1]:=Fekete; Szín[2..n]:=fehér
  Ciklus i=2-től n-ig
    Ha BeFok[i]=0 akkor
       x:=i; volt:=igaz
       Ciklus amíg x>1 és Szín[x]=fehér
          Ha volt akkor Szín[x]:=szürke
               különben M:=M+1; Szín[x]:=fekete
          x:=F[x]; Volt:=nem Volt
       Ciklus vége
       Ha Szín[x]≠fekete és nem Volt
          akkor M:=M+1; Szín[x]:=fekete
  Ciklus vége
Eljárás vége.
```

5. feladat: Csoport beosztás (15 pont)

Állítsuk elő a baráti kapcsolatok gráfja alapján a baráti komponenseket (lehet mélységi bejárással)!

```
Mélységi(gy,p):
  komp[p]:=gy
  Ciklus i=1-től Fok1[p]-ig
    q:=G1[p,i]
    Ha komp[q]=0 akkor Mélységi(gy,q)
  Ciklus vége
Eljárás vége.
```

Az egymást nem kedvelő párok esetén csak akkor lehet megoldás, ha a pár két tagja különböző komponensben van. Az ellenségeskedések gráfjában elég a két komponens közé húzni egy élt.

```
Csoportok:
  Ciklus i=1-től n-ig
    Ha komp[i]=0 akkor Mélységi(i,i)
  Ciklus vége
  Szín:=(fehér,…,fehér); Jó:=igaz
  Ciklus i=1-től n-ig
    x:=komp[i]
    Ha Szín[x]=fehér akkor
       Szín[x]:=piros; Sorüres; Sorba(x);
       Ciklus amíg nem üressor? és Jó
         p:=Sorból
         Ciklus y=1-től n-ig
           j:=komp[y]
           Ha x≠j és Szín[j]=fehér és G2[x,j] akkor
              Ha Szín[x]=Piros akkor Szín[j]:=kék
                            különben Szín[j]:=piros
              Sorba(j)
           különben ha x≠j és Szín[j]=Szín[x] és G2[x,j]
                       akkor Jó:=hamis
         Ciklus vége
       Ciklus vége
  Ciklus vége
Eljárás vége.
```

2012. Első forduló

Ötödik-nyolcadik osztályosok

1. feladat: Hibakeresés (24 pont)

Felismert hibánként 3-3 pont, összesen: 24 pont

A helyes algoritmusban vastagon szedettek és aláhúzottak az elrontott részek:

```
Keresztnevek(N,T,Db,K):
  Db:=1; K(Db):=T(1)      rossz a sorrend a feladatban,
                               de K(1):=T(1) is lehet a helyes
  Ciklus i=2-től N-ig
    j:=1
    Ciklus amíg j≤Db és T(i)≠K(j)
      j:=j+1
    Ciklus vége
    Ha j>Db akkor Db:=Db+1; K(Db):=T(i)
  Ciklus vége
Eljárás vége.
```

2. feladat: Mit csinál (26 pont)

A. A=5, K=1, V=3 — 1+1+1 pont

B. A=8, K=4, V=6 — 1+1+1 pont

C. A a legnagyobb összegű olyan összefüggő rész összege, amelyben csak nemnegatív számok vannak — 4 pont
kivéve, ha nincs ilyen, ekkor 0 0 0 a megoldás — 2 pont
K ennek a résznek a kezdete, — 2 pont
V pedig a vége — 2 pont
Ha több ilyen rész is van, akkor K és V az első rész kezdete és vége — 2 pont

D. A=0 marad, ha minden szám negatív vagy 0 — 2 pont

E. Ha a legnagyobb összegű rész vége az N. elem — 3 pont

F. Ha a T vektorban nincs negatív elem — 3 pont

3. feladat: Kitaláló (20 pont)

Értékelés: arányosan – lefelé kerekítve – részpont adható.

a a + +, a 0 + +, 0 a + +, a a + –, a 0 + –, 0 a + –, a a – +, a 0 – +, 0 a – + → A +	4 pont
a a – –, a 0 – –, 0 a – – → A –	2 pont
b b + +, b 0 + +, 0 b + +, b b –, b 0 + –, 0 b + –, b b – +, b 0 – +, 0 b – + → B +	4 pont
b b – –, b 0 – –, 0 b – – → B –	2 pont
a b + +, b a + +, a b + –, b a + –, a b – +, b a – + → A B +	3 pont
a b – –, b a – – → A B –	2 pont
0 0 + +, 0 0 + –, 0 0 – + → 0 +	2 pont
0 0 – – → 0 –	1 pont

Számítógépes feladat – VÁLASZTHATÓ

<u>4. feladat</u>: Időjárás (30 pont)

Három programozási tétel alkalmazásáról szól a feladat:

- megszámolás;
- keresés;
- maximumkiválasztás.

```
Napok száma(n,ma,mi,db):
  db:=0
  Ciklus i=1-től n-ig
    Ha ma[i]≥mi[i]+10 akkor db:=db+1
  Ciklus vége
Eljárás vége.
```

```
Napkeresés(n,ma,mi,van,sor):
  i:=1
  Ciklus amíg i<n és mi[i]≤ma[i+1]
    i:=i+1
  Ciklus vége
  van:=(i<n)
  Ha van akkor sor:=i különben sor:=0
Eljárás vége.
```

```
Legnagyobb eltérés(n,ma,mi,max):
  max:=1
  Ciklus i=1-től n-ig
    Ha ma[i]-mi[i]<ma[max]-mi[max] akkor max:=i
  Ciklus vége
Eljárás vége.
```

Számítógép nélküli feladat – VÁLASZTHATÓ

<u>4. feladat</u>: Időjárás (30 pont)

Minden helyes kitöltésre 5-5 pont

```
  Ha Max(i)>a akkor a:=Max(i)
  Ha (Max(i)+Min(i))/2<(Max(c)+Min(c))/2 akkor c:=i       *
  Ha Min(i)<0 és Max(i)>0 akkor volt:=igaz; b:=i          **
  Ha (Max(i)+Min(i))/2>Max(i-1) és
     (Max(i)+Min(i))/2<Min(i+1)
           akkor dbd:=dbd+1; sd:=sd+(Max(i)+Min(i))/2
  Ha dbd≠0 akkor d:=sd/dbd
```

*: a /2 elmaradhat a reláció mindkét oldaláról

**: az =0 is értelmezhető az olvadásnak

Kilencedik-tizedik osztályosok

1. feladat: Időjárás (30 pont)

A. Hibánként 3-3 pont adható. A helyes megoldásban aláhúzás jelöli az elrontott helyeket.

```
Forró(N,M,H,K,F):
  D:=0
  Ciklus i=1-től N-ig
    A:=0; j:=1
    Ciklus amíg j≤M és A<K
      Ha H(i,j)>F akkor A:=A+1 különben A:=0
      j:=j+1
    Ciklus vége
    Ha A≥K akkor D:=D+1; Y(D):=i
  Ciklus vége
Eljárás vége.
```

B. D a feltételnek megfelelő települések száma — 3 pont
az Y tömb 1..D eleme a megfelelő települések sorszáma — 3 pont

C. A az egymás után folyamatosan F fok feletti napok száma — 3 pont

2. feladat: Gyorsabbra (20 pont)

A részpontszámok 50%-a jár a magyarázatért, 50%-a pedig a megoldásért.

j<x helyett j≤gyök(x) lehet — 6 pont
részpont: j≤x/2 esetén 4 pont adható

A T(i)-k között lehetnek egyformák, ha felhasználja, hogy az adott számról korábban már kiderült, hogy prím-e, akkor — 8 pont
Részpont: ha előállítja a 2 és M közötti prímszámokat és a prímszámvizsgálatban azokkal oszt, akkor 4 pont adható.

Ha megoldja, hogy prím(T(i-1))-et már kiszámolta az előző lépésben — 2 pont

W:=igaz helyettesíthető az eljárás vége előtt Ha H>0 akkor W:=igaz utasítással — 4 pont

Egy gyors megoldás (memorizálás, gyors prímvizsgálat):

```
Szakasz(N,T,H,K,W):
  H:=0; V:=hamis; i:=1; W:=hamis; PR(1..M):=(hamis,...,hamis)
  Ha prím(T(1)) akkor Db:=1
  Ciklus i=2-től N-ig
    Ha PR(T(i)) akkor Ha PR(T(i-1)) akkor Db:=Db+1
                      különben Db:=1
    különben Ha prím(T(i)) akkor PR(T(i)):=igaz
                                 Ha PR(T(i-1)) akkor Db:=Db+1
                                 különben Db:=1
    különben Ha Db>H akkor H:=Db
  Ciklus vége
  Ha Db>H akkor H:=Db; W:=igaz
  Ha H>0 akkor W:=igaz
Eljárás vége.
```

```
prím(x):
  j:=2
  Ciklus amíg j≤gyök(x) és j nem osztója x
    j:=j+1
  Ciklus vége
  prím:=j>gyök(x)
Függvény vége.
```

3. feladat: Munka (25 pont)

A. haszon=40, munka sorszámok: 3,4,5,7,8	2+3 pont
B. haszon=25, munka sorszámok: 3,4,5,6,7	2+3 pont
C. haszon=25, munka sorszámok: 3,4,5,6,7	2+3 pont
D. haszon=31, munka sorszámok: 1,3,5,6,7	2+3 pont
E. haszon=45, munka sorszámok: 1,2,3,6,9	2+3 pont

4. feladat: Kitaláló (25 pont)

A.	T=(1,3,6,5,4,7,8,2)	2 pont
	T=(1,3,6,2,4,7,8,5)	2 pont
	T=(1,2,6,3,4,7,8,5)	2 pont
B.	T=(5,2,6,3,4,7,8)	2 pont
	T=(2,5,6,3,4,7,8)	2 pont
	T=(2,3,6,5,4,7,8)	2 pont
	x=1	2 pont
C.	T(i)≤T(2*i) és T(i)≤T(2*i+1)	3 pont
D.	T(i)≤T(2*i) és T(i)≤T(2*i+1)	3 pont
E.	j értéke a T(2*i) és a T(2*i+1) közül a kisebb indexe;	3 pont
	j értéke 2*i, ha 2*i=n	2 pont

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Időjárás (20 pont)

A. Hibánként az első 2 hibára 1-1, a továbbiakra 2-2 pont adható. A helyes megoldásban aláhúzás jelöli az elrontott helyeket.

```
Szélsőséges(N,M,H):
  C:=-1                   vagy C:=0; D:=0
  Ciklus i=1-től N-ig
    A:=H(i,1); B:=H(i,1)
    Ciklus j=2-től M-ig
      Ha H(i,j)>A akkor A:=H(i,j)
      Ha H(i,j)<B akkor B:=H(i,j)
    Ciklus vége
    Ha A-B>C akkor D:=1; Y(D):=i; C:=A-B
    különben ha A-B=C akkor D:=D+1; Y(D):=i
  Ciklus vége
Eljárás vége.
```

B. D a feltételnek megfelelő települések száma	2 pont
az Y tömb 1..D eleme a megfelelő települések sorszáma	2 pont
C. A az i-edik település maximális, B pedig a minimális hőmérséklete	2+2 pont

2. feladat: Gyorsabbra (20 pont)

A részpontszámok 50%-a jár a magyarázatért, 50%-a pedig a megoldásért.

j<x helyett j≤gyök(x) lehet — 4 pont
részpont: j≤x/2 esetén 2 pont adható

T(p) között lehetnek egyformák, ha felhasználja, hogy az adott számról korábban már kiderült, hogy prím-e, akkor — 6 pont
Részpont: ha előállítja a 2 és M közötti prímszámokat és a prímszámvizsgálatban azokkal oszt, akkor 4 pont adható.

van:=igaz helyettesíthető az eljárás vége előtt Ha D>0 akkor Van:=igaz utasítással 2 pont

A p-s ciklus felesleges (pontosabban a külső ciklus előtt egy H hosszú szakaszra ki kell számolni Db-t), a H hosszú szakaszok egymáshoz képest 2 elemben térnek el, a kilépő miatt Db eggyel csökken, ha az prím volt; a belépő miatt eggyel nő, ha prím volt 4+4 pont

Egy gyors megoldás (memorizálás, kumulatív összegzés, gyors prímvizsgálat):

```
Szakasz(N,T,H,K,Van):
  D:=0; Van:=hamis; PR(1..M):=(hamis,...,hamis); Db:=0
  Ciklus i=1-től H-ig
    Ha PR(T(i)) akkor Db:=Db+1
    különben ha prím(T(i)) akkor PR(T(i)):=igaz; Db:=Db+1
  Ciklus vége
  Ciklus i=2-től N-H+1-ig
    Ha PR(T(i-1)) akkor Db:=Db-1
    különben ha PR(T(i+H-1)) akkor Db:=Db+1
    különben ha prím(T(i+H-1)) akkor PR(T(i+H-1)):=igaz
                                     Db:=Db+1
    Ha Db>D akkor K:=i; D:=Db
  Ciklus vége
  Ha D>0 akkor Van:=igaz
Eljárás vége.
```

```
Prím(x):
  j:=2
  Ciklus amíg j≤gyök(x) és j nem osztója x
    j:=j+1
  Ciklus vége
  prím:=j>gyök(x)
Függvény vége.
```

3. feladat: Taxi (20 pont)

A. utasok száma: 7, utasok: 2,3,4,5,6,7,8 2+3 pont

B. utasok száma: 5, utasok: 2,3,4,5,6 2+3 pont

C. utasok száma: 8, utasok: 1,4,5,7,9,10,11,12 2+3 pont

D. utasok száma: 10, utasok: 1,3,4,5,7,8,9,10,11,12 2+3 pont

4. feladat: Kitaláló (20 pont)

A. T=(1,3,6,5,4,7,8,2) 2 pont
T=(1,3,6,2,4,7,8,5) 2 pont
T=(1,2,6,3,4,7,8,5) 2 pont

B. T=(5,2,6,3,4,7,8) 2 pont
x=1 1 pont

C. T(i)≤T(2*i) és T(i)≤T(2*i+1) 3 pont

D. Ha T(N)≥min(T(2),T(3)) biztos igaz volt, tehát T(N) T(1) helyére kerül és emiatt nem T(1) lesz a legkisebb), minden más elemre igaz marad a T(i)≤T(2*i) és T(i)≤T(2*i+1) 2+1 pont

E. j értéke a T(2*i) és a T(2*i+1) közül a kisebb indexe; 3 pont
j értéke 2*i, ha 2*i=n 2 pont

5. feladat: Képátló (20 pont)

A. K=3 3 pont

B. K=1 3 pont

C. K=3 3 pont

D. $T[i,j] = \min\left(T[i+1,j] + S[i,j+1], T[i,j+1] + O[i+1,j]\right)$, ha i<N és j<N 4 pont

T[i,j]=0, ha i=N vagy j=N 1 pont

ahol S[i,j]= az i-edik sorban a j–től jobbra levő 0-k száma 3 pont

O[i,j]= a j-edik oszlopban az i–től lefelé levő 1-esek száma 3 pont

2012. Második forduló

Ötödik-nyolcadik osztályosok

<u>1. feladat</u>: Időjárás (30 pont)

Az első két feladat egyszerű programozási tétel:

- maximumkiválasztás;
- megszámolás

A harmadik pedig leghosszabb adott tulajdonságú intervallum keresése.

```
Legmelegebb nap(n,h,max):
  max:=1
  Ciklus i=2-től n-ig
    Ha h[i]>h[max] akkor max:=i
  Ciklus vége
Eljárás vége.
```

```
Melegebb napok(n,h,db):
  db:=0
  Ciklus i=1-től n-ig
    Ha h[i]>k akkor db:=db+1
  Ciklus vége
Eljárás vége.
```

```
Leghosszabb időszak(n,h,kezdet,vég):
  h[0]:=-∞; h[n+1]:=-∞
  kezdet:=0; vég:=-1
  Ciklus i=1-től n-ig
    Ha h[i]>k és h[i-1]≤k akkor j:=i
    Ha h[i]>k és h[i+1]≤k
       akkor Ha i-j>vég-kezdet akkor kezdet:=j; vég:=i
  Ciklus vége
Eljárás vége.
```

<u>2. feladat</u>: Tükörkép számok (16 pont)

Ki kell szűrni a 0-val kezdődőket és a 0-val végződők tükörképeit!

```
Tükörszámok:
  Ciklus a=1-től 9-ig
    Ciklus b=1-től 9-ig
      s1:=a*b; s:=s1 mod 10
      r1:=(b*b+a*a+s1 div 10)
      q1:=(b*a+r1 div 10); q:=q1 mod 10
      p:=q1 div 10
      Ha p=0 és q=s akkor Ki: a,b,(a*10+b)*(a+b*10)
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

<u>3. feladat</u>: Életjáték (29 pont)

Egy kört játszunk le, két mátrixot használva (a kör elején és végén levő állapot). A sejtek helyén X, a többi helyen szóköz van a mátrixokban. Érdemes a szomszédszámolás egyszerűsége miatt a mátrixot körbevenni üres mezőket tartalmazó sorokkal és oszlopokkal. A további egyszerűsítés miatt a szomszédokba beleszámoljuk azt is, akinek a szomszédait számoljuk.

```
Játék(n,m,eleje,vége):
  vége:=eleje
  Ciklus i=1-tőn n-ig
    Ciklus j=1-től m-ig
      db:=0
      Ciklus k=i-1-től i+1-ig
        Ciklus l=j-1-től j+1-ig
          Ha eleje[k,l]=’X’ akkor db:=db+1
        Ciklus vége
      Ciklus vége
      Ha eleje[i,j]=’X’ akkor
         Ha db<3 vagy db>4 akkor vége[i,j]:=’ ’
      különben
         Ha db=3 akkor vége[i,j]:=’X’
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Előrejelzés (16 pont)

A feladatban egy mátrixra kell 4 különböző szempont szerint maximális sort meghatározni.

```
Első(n,m,h,max):
  maxért:=-∞
  Ciklus i=1-től n-ig
    Ciklus j=1-től m-ig
      Ha h[i,j]>maxért akkor max:=i; maxért:=h[i,j]
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

```
Második(n,m,h,max):
  maxért:=-∞
  Ciklus i=1-től n-ig
    s:=0
    Ciklus j=1-től m-ig
      s:=s+h[i,j]
    Ciklus vége
    Ha s>maxért akkor max:=i; maxért:=s
  Ciklus vége
Eljárás vége.
```

```
Harmadik(n,m,h,max):
  maxért:=-1; max:=-1
  Ciklus i=1-től n-ig
    h[i,0]:=-∞; h[i,m+1]:=-∞; kezdet:=0; vég:=-1
    Ciklus j=1-től m-ig
      Ha h[i,j]>k és h[i,j-1]≤k akkor c:=j
      Ha h[i,j]>k és h[i,j+1]≤k akkor
         Ha j-c>vég-kezdet akkor kezdet:=c; vég:=j
    Ciklus vége
    Ha vég-kezdet>maxért akkor max:=i; maxért:=vég-kezdet
  Ciklus vége
Eljárás vége.
```

```
Negyedik(n,m,h,max):
  dbmax:=(0,…,0)
  Ciklus j=1-től m-ig
    d1:=- ∞; d2:=hamis
    Ciklus i=1-től n-ig
      Ha h[i,j]>d1 akkor d1:=h[i,j]; d2:=igaz; d3:=i
                különben ha h[i,j]=d1 akkor d2:=hamis
    Ciklus vége
    Ha d2 akkor dbmax[d3]:=dbmax[d3]+1
  Ciklus vége
  max:=-1; maxért:=0
  Ciklus i=1-től n-ig
    Ha dbmax[i]>maxért akkor max:=i; maxért:=dbmax[i]
  Ciklus vége
Eljárás vége.
```

2. feladat: Verseny (20 pont)

Érdemes a részfeladatok megoldása előtt minden csapatra kiszámolni a győzelmek és vereségek számát, valamint annak a csapatnak a sorszámát, amely legyőzte (ilyen legfeljebb 1 lehet).

Az első részfeladat azok megadása, amelyek még nem kaptak ki, a második azoké, amelyek győztek is és kikaptak is. A harmadikban le kell számolni mindenkire, hogy hány csapatot győzött le közvetve, majd ezek maximumát venni!

```
Verseny(n,m,ered):
  gy:=(0,…,0); v:= (0,…,0); db:=(0,…,0); le:= (0,…,0)
  Ciklus i=1-től m-ig
    gy[ered[i,1]]:=gy[ered[i,1]]
    v[ered[i,2]]:=v[ered[i,2]]
    le[ered[i,2]]:=ered[i,1]
  Ciklus vége
  dba:=0
  Ciklus i=1-től n-ig
    Ha v[i]=0 akkor adb:=adb+1; a[adb]:=i
  Ciklus vége
  bdb:=0
  Ciklus i=1-től n-ig
    Ha gy[i]*v[i]>0 akkor bdb:=bdb+1; b[bdb]:=i
  Ciklus vége
  c:=0
  Ciklus i=1-től n-ig
    j:=i; k:=le[j]
    Ciklus amíg k>0
      db[k]:=db[k]+1
      Ha db[k]>db[c] akkor c:=k
      j:=k; k:=le[j]
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

<u>3. feladat</u>: Könyvtári pakolás (18 pont)

A megoldásban a ciklusokat kell felfedezni: a megoldás az elemek számából levonva a ciklusok számát.

```
Pakolás(n,a,db):
  m:=0
  Ciklus i=1-től n-ig
    Ha a[i]>0 akkor m:=m+1; y:=i; x:=a[y]; a[y]:=0
                    Ciklus amíg a[x]>0
                      y:=x; x:=a[y]; a[y]:=0
                    Ciklus vége
  Ciklus vége
  db:=n-m
Eljárás vége.
```

<u>4. feladat</u>: Számjáték (21 pont)

Dinamikus programozásos megoldást készítünk. Kiszámítjuk, hogy 1, 2, K lépés alatt mennyi az optimális megoldás értéke az első i elemet használva.

```
Számjáték(n,k,A,mego):
  Opt[0]:=0
  Ciklus j=1-től k-ig
    Ciklus i=N-től 2-ig -1-esével
      l:=(j-1)*2
      Ciklus amíg l≤i-2
        Ha Opt[l]+A[i-1]+A[i]>Opt[i]
           akkor Opt[i]:=Opt[l]+A[i-1]+A[i]
        l:=l+1
      Ciklus vége
    Ciklus vége
  Ciklus vége
  mego:=0
  Ciklus i=1-től n-ig
    Ha mego<Opt[i] akkor mego:=Opt[i]
  Ciklus vége
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

<u>1. feladat</u>: Előrejelzés (15 pont)

A feladatban egy mátrixra kell 4 különböző szempont szerint maximális sort meghatározni.

```
Első(n,m,h,max):
  maxért:=-1
  Ciklus i=1-től n-ig
    amax:=h[i,1]; amin:=h[i,1]
    Ciklus j=1-től m-ig
      Ha h[i,j]>amax akkor amax:=h[i,j]
      Ha h[i,j]<amin akkor amin:=h[i,j]
    Ciklus vége
    Ha amax-amin>maxért akkor max:=i; maxért:=amax-amin
  Ciklus vége
Eljárás vége.
```

```
Második(n,m,h,max):
  maxért:=-∞, i:=1; max:=-1
  Ciklus amíg i≤n és max=-1
    ii:=1
    Ciklus amíg ii≤n és b=-1
      Ha i≠ii akkor j:=1
                    Ciklus amíg j≤m és h[i,j]>h[ii,j]
                      j:=j+1
                    Ciklus vége
                    Ha j>m akkor b:=i különben ii:=ii+1
      különben ii:=ii+1
    Ciklus vége
    i:=i+1
  Ciklus vége
Eljárás vége.
```

```
Harmadik(n,m,h,max):
  maxért:=-1; max:=-1
  Ciklus i=1-től n-ig
    h[i,0]:=-∞; h[i,m+1]:=-∞; kezdet:=0; vég:=-1
    Ciklus j=1-től m-ig
      Ha h[i,j]>k és h[i,j-1]≤k akkor c:=j
      Ha h[i,j]>k és h[i,j+1]≤k akkor
         Ha j-c>vég-kezdet akkor kezdet:=c; vég:=j
    Ciklus vége
    Ha vég-kezdet>maxért akkor max:=i; maxért:=vég-kezdet
  Ciklus vége
Eljárás vége.
```

```
Negyedik(n,m,h,max):
  dbmax:=(0,…,0)
  Ciklus j=1-től m-ig
    d1:=- ∞; d2:=hamis
    Ciklus i=1-től n-ig
      Ha h[i,j]>d1 akkor d1:=h[i,j]; d2:=igaz; d3:=i
                 különben ha h[i,j]=d1 akkor d2:=hamis
    Ciklus vége
    Ha d2 akkor dbmax[d3]:=dbmax[d3]+1
  Ciklus vége
  max:=-1; maxért:=0
  Ciklus i=1-től n-ig
    Ha dbmax[i]>maxért akkor max:=i; maxért:=dbmax[i]
  Ciklus vége
Eljárás vége.
```

<u>2. feladat</u>: Verseny (15 pont)

Érdemes a részfeladatok megoldása előtt minden csapatra kiszámolni a győzelmek és vereségek számát, valamint annak a csapatnak a sorszámát, amely legyőzte (ilyen legfeljebb 1 lehet).

Az első részfeladat egy olyan csapat megadása, már kikapott, de a legtöbbször győzött. A másodikban le kell számolni mindenkire, hogy hány csapatot győzött le közvetve, majd ezek maximumát venni! A harmadikban az előzőben kiszámolt győzelmek alapján a két legkisebbet kell megadni.

```
Verseny(n,m,ered,a,b):
  gy:=(0,…,0); v:= (0,…,0); db:=(0,…,0); le:= (0,…,0)
  Ciklus i=1-től m-ig
    gy[ered[i,1]]:=gy[ered[i,1]]
    v[ered[i,2]]:=v[ered[i,2]]
    le[ered[i,2]]:=ered[i,1]
  Ciklus vége
  maxért:=0; a:=-1
  Ciklus i=1-től n-ig
    Ha v[i]>0 akkor
       Ha gy[i]>maxért akkor a:=i; maxért:=gy[i]
  Ciklus vége
  b:=0
  Ciklus i=1-től n-ig
    j:=i; k:=le[j]
    Ciklus amíg k>0
      db[k]:=db[k]+1
      Ha db[k]>db[b] akkor b:=k
      j:=k; k:=le[j]
    Ciklus vége
  Ciklus vége
  Ha m=n-1 akkor c1:=-1; c2:=-1
  különben c1:=0; c2:=0; db[0]:=+∞
           Ciklus i=1-től n-ig
             Ha v[i]=0 akkor
                Ha db[i]<db[c1] akkor c2:=c1; c1:=i
                különben ha db[i]<db[c2] akkor c2:=i
           Ciklus vége
Eljárás vége.
```

<u>3. feladat</u>: Üvegválogatás (15 pont)

Meg kell nézni az üvegek összes permutációját, s közülük kiválasztani azt, amihez a legkevesebb üveget kell mozgatni!

```
Válogat(n,L,mego):
  ip:=(igaz,…,igaz); s:=0
  Ciklus i=1-től n-ig
    Ciklus j=1-től n-ig
      s:=s+L[i,j]
    Ciklus vége
  Ciklus vége
  Perm(1); mego:=s-M
Eljárás vége.
```

```
Perm(i):
  Ciklus x=1-től n-ig
    Ha ip[x] akkor P[i]:=x; ip[x]:=hamis
                   Ha i=n akkor VM:=0
                                Ciklus j=1-től n-ig
                                  VM:=VM+L[i,P[i]])
                                Ciklus vége
                                Ha VM>M akkor M:=VM; P0:=P
                   különben Perm(i+1)
    ip[x]:=igaz
  Ciklus vége
Eljárás vége
```

4. feladat: Párosítás (15 pont)

Ha az `i`. helyen az `a[i]` érték van, akkor az `a[i]`. helyen az `i` értéknek kellene lenni!

```
Párosítás(n,a,s):
  Ciklus i=1-től n-ig
    c[a[i]]=i
  Ciklus vége
  s:=0;
  Ciklus i=1-től n-ig
    Ha a[i]≠i akkor a[c[i]]=a[i]; c[a[i]]=c[i]; s:=s+1
  Ciklus vége
Eljárás vége.
```

5. feladat: Számjáték (15 pont)

A feladat az előző korcsoportos feladat nehezítése.

```
Számjáték(n,A,H,L):
  Szum:=(0,…,0); x:=0
  Ciklus i=1-től H-ig
    x:=x+A[i]
  Ciklus vége
  Szum[H]:=x
  Ciklus x=H+1-től n-ig
    Szum[x]:=Szum[x-1]+A[x]-A[x-H]
  Ciklus vége
  Opt[0,0..N]:=(0,…,0)
  Ciklus ll=1-től L-ig
    Ciklus x=H-tól N-ig
      opti:=Opt[ll-1,x]
      Ha Opt[ll,x-1]>opti akkor opti:=Opt[ll,x-1]
      Ha Opt[ll-1,x-H]+Szum[x]>opti
         akkor opti:=Opt[ll-1,x-H]+Szum[x]
      Opt[ll,x]:=opti
    Ciklus vége
  Ciklus vége
  Ki: Opt[L,N])
  ll:=L; x:=N
  Ciklus amíg ll>0 és x≥H
    Ciklus amíg x>0 és ll>0 és Opt[ll,x]=Opt[ll,x-1]
      x:=x-1
    Ciklus vége
    Ki: x-H+1; x:=x-H; ll:=ll-1
  Ciklus vége
Eljárás vége.
```

2012. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Állat (25 pont)

A feladat megoldása egy megszámolás (melyik állat hány út végén található), majd erre a tömbre egy megszámolás (hány 1-es van benne) és egy maximumkiválasztás (mi a legnagyobb értéke).

```
Állat(n,m,a,b,db,max):
  d:=(0,…,0)
  Ciklus i=1-től m-ig
    Ha a[i]>0 akkor d[a[i]]:=d[a[i]]+1
    Ha b[i]>0 akkor d[b[i]]:=d[b[i]]+1
  Ciklus vége
  db:=0; max:=1
  Ciklus i=1-től n-ig
    Ha d[i]=1 akkor db:=db+1
    Ha d[i]>d[max] akkor max:=i
  Ciklus vége
Eljárás vége.
```

2. feladat: Titkosírás (25 pont)

Szükségünk lesz egy konstans mátrixra a kódoló táblázathoz, sorait 'a'-tól 'c'-ig, oszlopait 'a'-tól 'i'-ig indexeljük:

```
  t =(('i','o','q','h','b','f','y','l','w'),
      ('n','r','a','g','s','k','t','e','z'),
      ('d','u','p','x','c','j','v','m',' '))
```

A titkosítás algoritmusa a táblázat belsejében keres, majd megadja megtalált betű két indexét.

```
Titkosítás(a,b):
  b:=''
  Ciklus i=1-től hossz(a)-ig
    j:='a'; k:='a'
    Ciklus amíg t[j,k]≠a[i]
      Ha k='i' akkor j:=következő(j); k:='a'
            különben k:=következő(k)
    Ciklus vége
    b:=b+j+k
  Ciklus vége
Eljárás vége.
```

A visszafejtés egyszerűbb, a két szomszédos betűt a táblázathoz indexnek használva, megkapjuk a kódolt betűt:

```
Visszafejtés(b,a):
  a:=''; i:=1
  Ciklus amíg i<hossz(b)
    a:=a+t[b[i],b[i+1]]; i:=i+2
  Ciklus vége
Eljárás vége.
```

Ha egy programozási nyelvben nem lehet karakterekkel indexelni, akkor a karaktereket át kell kódolni egész számokká!

3. feladat: Alma (25 pont)

A feladat két multihalmaz metszetéből számítja ki a megoldást. A metszetbeli értéket levonja a termelőtől, a kereskedő mennyiségét pedig átírja a metszetbeli értékre. Ezzel a termelő multihalmazában azok maradnak, amit nem tudott eladni, a kereskedőében pedig azok, amelyeket meg tudott venni. A végeredményben a két multihalmaz azon elemeit kell kiírni, amelyek darabszáma nem 0!

```
Alma(n,t,m,k):
  Ciklus j=1-től m-ig
    i:=1
    Ciklus amíg i≤n és t[i].fajta≠k[j].fajta
      i:=i+1
    Ciklus vége
    Ha i≤n akkor
       Ha k[j].menny>t[i].menny
          akkor k[j].menny:=t[i].menny; t[i].menny:=0
       különben t[i].menny:=t[i].menny-k[j].menny
    különben k[j].menny:=0
  Ciklus vége
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Állatkert (15 pont)

A feladat egy gráf szélességi bejárása, ahol számoljuk az egyes pontokba vezető legrövidebb utak számát. Ha új pontba érünk, akkor ez annyi, amennyi az adott él kezdőpontja legrövidebb útjai száma, ha olyanba, ahol már voltunk és szintén legrövidebb utat találtunk, akkor a két darabaszámot össze kell adni! A bejárat a 0-s pont, onnan indulunk.

```
Legrövidebb utak(n,fok,g,db)
  szín:=(fehér,…,fehér)
  Sorüres; Sorba(0); szín[0]:=szürke; táv[0]:=0
  Ciklus amíg nem üressor?
    Sorból(i)
    Ciklus j∊g[i]
      Ha szín[j]=fehér akkor Sorba(j); szín[j]:=szürke
                             db[j]:=db[i]; táv[j]:=tav[i]+1
      különben ha szín[j]=szürke és táv[j]=táv[i]+1
                   akkor db[j]:=db[j]+db[i]
    Ciklus vége
    szín[i]:=fekete
  Ciklus vége
Eljárás vége.
```

2. feladat: Róka (15 pont)

Ez egy összetett Fibonacci-szám variáció. A d vektor első L elemében az előző L évben született rókák számát tároljuk. A vektor további elemeit kell számolni, majd a végén az utolsó L érték összege a végeredmény.

```
Rókák(n,L,K,db,ered):
  Ciklus i=1-től L-ig
    d[i]:=db[L-i+1]
  Ciklus vége
```

```
  Ciklus i=L+1-től L+n-ig
    d[i]:=0
    Ciklus j=i-1-től i-k-ig
      d[i]:=(d[i]+d[j]) mod 1000000
  Ciklus vége
  db:=0
  Ciklus i=n+1-től n+L-ig
    db:=(db+d[i]) mod 1000000
  Ciklus vége
Eljárás vége.
```

3. feladat: Gyöngy (15 pont)

A lánc első K elemét másoljuk le a lánc végére, így a feladat megoldásánál nem kell a ciklikussággal foglalkoznunk!

Ezután az K. és K+1. hely közötti vágáshoz számoljuk le, hogy tőle balra hány piros, illetve jobbra hány zöld gyöngy van! A vágási helyet léptetve ezek száma legfeljebb eggyel változik.

```
Gyöngyök(n,k,L,maxért,max):
  Ciklus i=1-től k-ig
    L[n+i]:=L[i]
  Ciklus vége
  b:=0; j:=0; maxért:=|b-j|; max:=k
  Ciklus i=1-től k-ig
    Ha L[i]='P' akkor b:=b+1
    Ha L[i+k]='Z' akkor j:=j+1
  Ciklus vége
   Ciklus i=k+1-től n-ig
    Ha L[i-k]='P' akkor b:=b-1
    Ha L[i]='P' akkor b:=b+1 különben j:=j-1
    Ha L[i+k]='Z' akkor j:=j+1
    Ha |b-j|<maxért akkor maxért:=|b-j|; max:=i
  Ciklus vége
Eljárás vége.
```

4. feladat: Raktár (15 pont)

Ez egy szélességi bejárás variáció, egy fát kell bejárni, a levelektől indulva (figyelni kell rá, hogy a raktár is lehet levél). Minden pontra számítsuk ki, hogy mennyi áru hozható ide a levelek felől, majd innen legfeljebb M-et engedjünk tovább vinni a raktár felé!

```
Raktár(fok,g):
  Sorüres
  Ciklus i=1-től n-ig
    Ha fok[i]=1 és i≠r akkor Sorba(i)
  Ciklus vége
  Ciklus amíg nem üressor?
    Sorból(i); j:=g[i].él[1]
    Ha a[i]<m akkor a[j]:=a[j]+a[i] különben a[j]:=a[j]+m
  {(i,j) él törlése j sorából}
    k:=1
    Ciklus amíg g[j].él[k]≠i
      k:=k+1
    Ciklus vége
    g[j].él[k]:=g[j].él[fok[j]];fok[j]:=fok[j]-1
  {ha j levél lett, akkor be a sorba}
    Ha fok[j]=1 és j≠r akkor Sorba(j)
  Ciklus vége
Eljárás vége.
```

5. feladat: Munka (15 pont)

Ez egy dinamikus programozásos feladat, az kell kiszámítani, hogy mi a legnagyobb haszon, ha az i. az utolsónak kiválasztott. Rendezzük sorba az adatainkat befejező nap szerint növekvő sorrendbe és vegyünk fel egy fiktív, 0. munkát!

```
Munka(n,kezdet,vég,érték,h)
  Rendez(n,kezdet,vég,érték)
  m[0]:=0; vég[0]:=0
  Ciklus i=1-től n-ig
    j:=1; k:=0; m[i]:=érték[i]
    Ciklus amíg vég[j]<kezdet[i]
      Ha m[j]+érték[i]>m[i] akkor m[i]:=m[j]+érték[i]
      j:=j+1
    Ciklus vége
  Ciklus vége
  h:=m[1]
  Ciklus i=2-től n-ig
    Ha m[i]>h akkor h:=m[i]
  Ciklus vége
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Szállítás (15 pont)

A feladat egy Dijkstra algoritmus variáció, azt kell megadni, hogy mi a legnagyobb súlyú él, amilyen élsúly korláttal van még út A-ból B-be.

```
Szállítás(n,fok,gr,a,b,ered):
  szín:=(szürke,...,szürke)
  súly:=(0,...,0); honnan[a]:=a; súly[a]:=+∞; PrSorÜres
  Ciklus i=1-től n-1-ig
    Prsorból(p); szín[p]:=fekete
    Ciklus k∈gr[p]
      Ha szín[k.hova]=szürke akkor
         Ha k.súly>súly[p] akkor q:=súly[p] különben q:=k.súly
         Ha q>súly[k] akkor súly[k]:=q; honnan[k]:=p
                            PrSorbanMozgat(k)
    Ciklus vége
  Ciklus vége
  ered:=súly[b]
Eljárás vége.
```

A kiírás a honnan vektor alapján visszafelé történik, klasszikus balrekurzióval:

```
Kiírás(a,b):
  Ha a=b akkor Ki: a
      különben Kiírás(a,honnan[b]); Ki: b
Eljárás vége.
```

2. feladat: Nyúl (15 pont)

Ez egy összetett Fibonacci-szám variáció. A `d` vektor első `L` elemében az előző `L` évben született rókák számát tároljuk. A vektor további elemeit kell számolni, majd a végén az utolsó L érték összege a végeredmény.

```
Nyúl(n,L,K,db,r,ered):
  Ciklus i=1-től L-ig
    d[i]:=db[L-i+1]
  Ciklus vége
  Ciklus i=L+1-től L+n-ig
    d[i]:=0
    Ciklus j=i-1-től i-k-ig
      d[i]:=(d[i]+d[j]*r[i-j]) mod 1000000
    Ciklus vége
  Ciklus vége
  db:=0
  Ciklus i=n+1-től n+L-ig
    db:=(db+d[i]) mod 1000000
  Ciklus vége
Eljárás vége.
```

<u>3. feladat</u>: Nyaklánc (15 pont)

Szükségünk lesz az N/2 hosszú szakaszok összegére (`sum`), az elvágási pozíciótól balra, illetve jobbra közelebb levő nyaklánc értékre (`bal`, `jobb`). A cél azon pozíció kiválasztása, amelyre `bal[i]+jobb[i]` maximális.

```
Nyaklánc(n,ertek,mego):
  sum[1]:=0
  Ciklus i=1-től n div 2-ig
    sum[1]:=sum[1]+ertek[i]
  Ciklus vége
  Ciklus i=2-től n-ig
    j:=i-1+n div 2
    Ha j>n akkor j:=j-n
    sum[i]:=sum[i-1]-ertek[i-1]+ertek[j]
  Ciklus vége
  sum[n+1]:=sum[1]
  bal[n div 2]:=0
  Ciklus i=1-től n/2-ig
    bal[n div 2]:=bal[n div 2]+ertek[i]*(n div 2 - i+1)
  Ciklus vége
  jobb[n div 2]:=0
  Ciklus i=n div 2 +1-től n-ig
    jobb[n div 2]:=jobb[n div 2]+ertek[i]*(i-n div 2)
  Ciklus vége
  max:=n div 2; i:=n div 2+1; j:=i-1; k:=1
  Ciklus amíg i≠n div 2
    bal[i]:=bal[j]-(n div 2)*ertek[k]+sum[k+1]
    jobb[i]:=jobb[j]-sum[i]+(n div 2)*ertek[k]
    Ha bal[i]+jobb[i]>bal[max]+jobb[max] akkor max:=i
    k:=k+1; j:=j+1
    Ha j>n akkor j:=1
    i:=i+1
    Ha i>n akkor i:=1
  Ciklus vége
Eljárás vége.
```

<u>4. feladat</u>: Üzletek (15 pont)

A sorban kezdetben a levelek vannak, kivéve esetleg az r pontot. A fagráfot a leveleitől kezdve dolgozzuk fel. Az a vektorba (amiben kezdetben az egyes településeken elérhető haszon van) azt számoljuk ki, hogy az egyes települések haszna mennyire változik, ha kell szállítani oda árut. Ha a szállítás többe kerülne, mint a haszon, akkor ott nem lesz üzlet. Ha nem kerül többe, akkor az elérhető hasznot adjuk hozzá ahhoz a településhez, ahonnan közvetlenül jön áru erre a településre. Így `a[i]` értéke már az lesz, hogy mekkora lehet a haszon az `i.` településtől az összes, a fában alatta levő településeken.

```
Üzletek(n,fok,a,u,volt):
  v:=u; fokv:=fok; Sorüres
  Ciklus i=1-től n-ig
    Ha fok[i]=1 és i≠r akkor Sorba(i)
  Ciklus vége
  Ciklus amíg nem üressor?
    Sorból(i); j:=u[i].hova[1]
    ha a[i]-u[i].súly[1]>0 akkor a[j]:=a[j]+a[i]-u[i].súly[1]
                          különben a[i]:=0
  {(i,j) él törlése j sorából}
    k:=1
    Ciklus amíg k<fok[j] és u[j].hova[k]≠i
      k:=k+1
    Ciklus vége
    u[j].hova[k]:=u[j].hova[fok[j]]
    u[j].súly[k]:=u[j].súly[fok[j]]
    fok[j]:=fok[j]-1
  {ha j levél lett, akkor be a sorba}
    Ha fok[j]=1 és j≠r akkor Sorba(j)
  Ciklus vége
  volt[i]:=(hamis,…,hamis)
  Sorüres; Sorba(r); volt[r]:=igaz
  Ciklus amíg nem üressor?
    Sorból(i)
    Ciklus j∊v[i].hova
      Ha a[j]>0 és nem volt[j] akkor Sorba(j); volt[j]:=igaz
    Ciklus vége
  Ciklus vége
Eljárás vége
```

<u>5. feladat</u>: Párok (15 pont)

Ez egy dinamikus programozásos feladat, számítsuk ki megoldást az első `i` darab 11. osztályos és első `j` darab 12. osztályos tanuló esetére! A gráf pontjait két csoportba soroljuk, `u`-ba a 11. osztályosok, `v`-be a 12. osztályosok kerüljenek! Az `u`-ból minden él `v`-be vezet, a `v`-ből pedig `u`-ba. Minden pontnál célszerű a kivezető éleket a másik végpontjuk szerint sorba rendezni, ekkor ugyanis, ha `(i,j)` él szerepel a megoldásban, akkor az összes további megoldásbeli `(k,l)` élekre igaz, hogy `k>i` és `l>j`.

Legyen `mego[i,j]` a megoldás az `(i,j)` párig, `el[i,j]` pedig a megoldás utolsó párosítása!

Ha az első 11. osztályosnak nincs ismerőse, vagy az első ismerőse `i`-nél nagyobb sorszámú, akkor az `(1,i)` párra nincs megoldás. Hasonló a tennivaló az `(i,1)` párokkal is.

```
Párok(n,foku,u,fokv,v,ered,mego,el)
  Ciklus i=1-től n-ig
    ha foku[1]=0 vagy i<u[1,1] akkor mego[1,i]:=0
    különben mego[1,i]:=1; el[1,i]:=(1,u[1,1])
  Ciklus vége
  Ciklus i=1-től n-ig
    ha fokv[1]=0 vagy i<v[1,1] akkor mego[i,1]:=0
    különben mego[i,1]:=1; el[i,1]:=(v[1,1],1)
  Ciklus vége
```

```
  Ciklus i=2-től n-ig
    Ciklus j=2-től n-ig
  {ha van i,j él, akkor mego(i-1,j-1)+1 is lehet}
      k:=1
      Ciklus amíg k≤foku[i] és u[i,k]<j
        k:=k+1
      Ciklus vége
      Ha u[i,k]=j akkor mego[i,j]:=mego[i-1,j-1]+1
                        el[i,j]:=(i,j)
               különben mego[i,j]:=mego[i-1,j-1]
                        el[i,j]:=el[i-1,j-1]
  {i,akármi pár}
      k:=1
      Ciklus amíg k≤foku[i] és u[i,k]<j
        Ha mego[i,j]<mego[i,u[i,k]
           akkor mego[i,j]:=mego[i,u[i,k]]
                 el[i,j]:=el[i,u[i,k]]
        k:=k+1
      Ciklus vége
  {akármi,j pár}
      k:=1
      Ciklus amíg k≤fokv[j] és v[j,k]<i
        Ha mego[i,j]<mego[v[j,k],j]
           akkor mego[i,j]:=mego[v[j,k],j]
                 el[i,j]:=el[v[j,k],j]
        k:=k+1
      Ciklus vége
    Ciklus vége
  Ciklus vége
  ered:=mego[n,n]
Eljárás vége.
```

A kiírás eljárás az utolsó kiírandó élt kapja paraméterül, az eljárásban használunk párokra definiált műveleteket.

```
kiiras((i,j)):
  Ha el[i,j]≠(0,0)
     akkor Ha el[i,j=(i,j) akkor kiiras(el[i,j]-(1,1))
                                 Ki: el[i,j]
           különben kiiras(el[i,j])
Eljárás vége.
```

2013. Első forduló

Ötödik-nyolcadik osztályosok

Számítógép nélküli feladatok

1. feladat: Karesz a robot (44 pont)

A. (10,10) 5 pont

B. (11,10) 5 pont

C. Ha X<Y, akkor a (10,10) mezőn 4 pont
Ha X=Y, akkor a (10,10) mezőn 1 pont
Ha X>Y, akkor a (11,10) mezőn 5 pont

D. Ha X≤Y: akkor a (10,11) mezőn: X, a (11,10) mezőn: Y-X, a (11,9) mezőn: X 4+4+4 pont
(ha bármely más mezőre nem 0-t mond, akkor 4 pont levonandó)
Ha X>Y: akkor a (10,10) mezőn:X-Y-1, a (10,11) mezőn: Y+1, a (11,9) mezőn: Y 4+4+4 pont
(ha bármely más mezőre nem 0-t mond, akkor 4 pont levonandó)

2. feladat: Mit csinál (44 pont)

A D..G részfeladatokra az alábbival ekvivalens megfogalmazások is elfogadandók:

A. M=1; Y(1)=2 2+2 pont

B. M=2; Y=(2;3) 2+2+2 pont

C. M=3; Y=(1;2;3) 2+2+2+2 pont

D. A legnagyobb számból egyetlen van X-ben 5 pont

E. Az összes szám egyforma 5 pont

F. M: az első i szám között hány maximális értékű szám van; Y az első i között a maximálisak sorszámai 4+4 pont

G. M: az X-ben hány maximális értékű szám van; Y az X-ben a maximálisak sorszámai 4+4 pont

3. feladat: Hegyek-völgyek (52 pont)

A C, D és E részre ugyanezt jelentő más megfogalmazások is elfogadhatók:

A. L,H,K,V 2+2+2+2 pont

B. L=20; H=12; K=17; V=28 4+4+4+4 pont

C. L a legkisebb érték helye; H a legszélesebb völgy hossza; K a legszélesebb völgy kezdete; V a legszélesebb völgy vége 4+4+4+4 pont

D. VH az aktuális völgy hossza; VK az aktuális völgy kezdete; VV az aktuális völgy vége 3+3+3 pont

E. Ha nincs völgy (azaz X(1)-től kezdődően valameddig monoton növekvő a sorozat, onnan a végéig pedig monoton csökkenő) 3 pont

Számítógépes feladat – VÁLASZTHATÓ

4. feladat: Lakások (60 pont)

Az első két feladat egyszerű programozási tétel alkalmazása: maximumkiválasztás, illetve megszámolás.

```
Legdrágább(n,ter,ár,max):
  max:=1
  Ciklus i=2-től N-ig
    Ha ár[i]>ár[max] akkor max:=i
  Ciklus vége
Eljárás vége.
```

```
Nagy és olcsó(n,ter,ár,db):
  db:=0
  Ciklus i=1-től n-ig
    Ha ter[i]>100 és ár[i]<40 akkor db:=db+1
  Ciklus vége
Eljárás vége.
```

A harmadik feladat kétféleképpen is megoldható:

- számoljuk minden alapterület első előfordulását
- tároljuk egy logikai tömbben, hogy milyen alapterületű lakások vannak (ehhez tudni kell, hogy mekkora a legnagyobb lakás).

```
Hányféle-1(n,ter,ár,db):
  db:=1
  Ciklus i=2-től n-ig
    j:=1
    Ciklus amíg ter[i]≠ter[j]
      j:=j+1
    Ciklus vége
    Ha i=j akkor db:=db+1
  Ciklus vége
Eljárás vége.
```

```
Hányféle-2(n,ter,ár,db):
  log:=(hamis,…,hamis)
  Ciklus i=1-től n-ig
    log[ter[i]]:=igaz
  Ciklus vége
  db:=0
  Ciklus i=1-től maxméret-ig
    Ha log[i] akkor db:=db+1
  Ciklus vége
Eljárás vége.
```

Számítógép nélküli feladat – VÁLASZTHATÓ

4. feladat: Lakások (60 pont)

A D és S szerepe felcserélhető, a megoldást úgy is el kell fogadni:

```
Lakás(N,T,A,M,B,L,X):
  M:=1; D:=0; S:=0; L:=0
  Ciklus i=1-től N-ig
    Ha T(i)≤T(M) akkor M:=i                          6+6 pont
    Ha T(i)<100 akkor D:=D+1; S:=S+A(i)              6+6+6 pont
    Ha A(i)>100000000 akkor L:=L+1; X(L):=i          6+6+6 pont
  Ciklus vége
  Ha D>0 akkor B:=S/D különben B:=0                  6+6 pont
Eljárás vége.
```

Ha a T(i)≤T(M) helyére T(i)<T(M)-t ír, akkor az első 6 pont helyett csak 5 adható.

Kilencedik-tizedik osztályosok

1. feladat: Karesz a robot (44 pont)

A. (10,10) 5 pont

B. (11,10) 5 pont

C. Ha X<Y, akkor a (10,10) mezőn 4 pont
Ha X=Y, akkor a (10,10) mezőn 1 pont
Ha X>Y, akkor a (11,10) mezőn 5 pont

D. Ha X≤Y: akkor a (10,11) mezőn: X, a (11,10) mezőn: Y-X, a (11,11) mezőn: X 4+4+4 pont
(ha bármely más mezőre nem 0-t mond, akkor 4 pont levonandó)
Ha X>Y: akkor a (10,10) mezőn:X-Y-1, a (10,11) mezőn: Y+1, a (11,11) mezőn: Y 4+4+4 pont
(ha bármely más mezőre nem 0-t mond, akkor 4 pont levonandó)

2. feladat: Mit csinál (54 pont)

A D..I részfeladatokra az alábbival ekvivalens megfogalmazások is elfogadandók:

A. M=1; Y(1)=3 2+2 pont

B. M=2; Y=(2,3) 2+2+2 pont

C. M=3; Y=(1,2,3) 2+2+2+2 pont

D. M=0, ha nincs páros szám X-ben 4 pont

E. A legnagyobb páros számból egyetlen van X-ben 5 pont

F. Az összes szám páros és egyforma 5 pont

G. M: az első i szám között hány maximális értékű páros szám van; Y az első i között a maximális párosak sorszámai 5+5 pont

H. M: az X-ben hány maximális értékű páros szám van; Y az X-ben a maximális párosak sorszámai 4+4 pont

I. K: a legnagyobb páros szám i-ig 4 pont

3. feladat: Hegyek-völgyek (52 pont)

A C, D és E részre ugyanezt jelentő más megfogalmazások is elfogadhatók:

A. M,H,K,V 2+2+2+2 pont

B. M=21; H=12; K=17; V=28 4+4+4+4 pont

C. M a völgyek alacsonyabb oldalai magasságának a maximuma; H a legszélesebb völgy hossza; K a legszélesebb völgy kezdete; V a legszélesebb völgy vége 4+4+4+4 pont

D. VH völgy hossza; VK az aktuális völgy kezdete; VV az aktuális völgy vége 3+3+3 pont

E. Ha nincs völgy (azaz X(1)-től kezdődően valameddig monoton növekvő a sorozat, onnan a végéig pedig monoton csökkenő) 3 pont

4. feladat: Raktár (50 pont)

A. AABABB vagy BABAAB (bármelyik elfogadható) 6*2 pont
minimális költség: 13 6 pont

B. Költség(0,0)=0 2 pont
Költség(i,0)=Költség(i-1,0)+A(i) 6 pont

Költség(0,j)=Költség(0,j-1)+B(j) 6 pont

Költség(i,j)=min(6 pont
Költség(i-1,j)+A(i+j), 6 pont
Költség(i,j-1)+B(i+j)) 6 pont

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Karesz a robot (40 pont)

A nem változásért járó pont akkor nem adható meg, ha a versenyző változást írt:

A. (10,16) 5 pont

B. (10,17) 5 pont

C. (10,(X+Y) div 2) 10 pont

D. (10,(X+Y) div 2) mezőn 1 kavics; a többi helyen nem lesz kavics; Karesz kezében Z+1 kavics lesz 3+3+3 pont

F. Z>=Y-X-1 (azaz a (10,X) és a (10,Y) mezők között legfeljebb Z üres hely van) 5 pont

F. Karesz ugyanott állna meg; a (10,X) és (10,Y) mezők között mindenhol lenne egy-egy kavics
2+4 pont

2. feladat: Mit csinál (46 pont)

Az E..I részfeladatokra az alábbival ekvivalens megfogalmazások is elfogadandók:

A. P=1; Q=1; Y(1)=3; Z(1)=1 1+1+1+1 pont

B. P=2; Q=2; Y=(2,4); Z=(1,3) 1+1+1+1 pont

C. P=0, ha nincs páros szám X-ben 3 pont

D. Q=0, ha nincs páratlan szám X-ben 3 pont

E. A legnagyobb páros számból egyetlen van X-ben 5 pont

F. Az összes szám páratlan és egyforma 5 pont

G. P az első i szám között hány maximális értékű páros szám van; Q az első i szám között hány minimális értékű páratlan szám van; Y az első i között a maximális párosak sorszámai; Z az első i között a minimális páratlanok sorszámai 2+2+2+2 pont

H. P az X-ben hány maximális értékű páros szám van; Q az X-ben hány minimális értékű páratlan szám van; Y az X-ben a maximális párosak sorszámai; Z az X-ben a minimális páratlanok sorszámai 2+2+2+2 pont

I. K: a legnagyobb páros szám i-ig; L: a legkisebb páratlan szám i-ig 3+3 pont

3. feladat: Hegyek-völgyek (44 pont)

A C, D, E és F részre ugyanezt jelentő más megfogalmazások is elfogadhatók:

A. M,H,K,V — 1+1+1+1 pont

B. M=7; H=12; K=17; V=28 — 3+3+3+3 pont

C. M a völgyek alacsonyabb oldalai magassága és az alja magassága különbségeinek a maximuma; H a legszélesebb völgy hossza; K a legszélesebb völgy kezdete; V a legszélesebb völgy vége
3+3+3+3 pont

D. VH völgy hossza; VK az aktuális völgy kezdete; VV az aktuális völgy vége; VA az aktuális völgy legalacsonyabb pontja — 2+2+2+2 pont

E. Ha nincs völgy (azaz X(1)-től kezdődően valameddig monoton növekvő a sorozat, onnan a végéig pedig monoton csökkenő) — 4 pont

F. A kis mélységű völgyeket és hegyeket eltünteti — 4 pont

4. feladat: Anyagvizsgálat (35 pont)

A. 15 — 6 pont

B. 5 vagy 9 bármelyike jó — 6 pont

C. $(A+1)^{U}-1$ — 7 pont
6 pont, ha a -1 lemarad

D. $(A+1)^{U}(B+1)^{V}-1$ — 8 pont
6 pont, ha a -1 lemarad

E. $(A+1)^{(U-1)}(B+1)^{V}+1$ vagy $(A+1)^{U}(B+1)^{(V-1)}+1$ — 8 pont
6 pont, ha a +1 lemarad

5. feladat: Munkavállalás (35 pont)

A. 1. megoldás: Nap(j)=i, ha a j. napon az i. munkát kell elvégezni, vagy 0 — 5+1 pont
2. megoldás: Nap(j)=i, ha a j. napon az i. munkát kell elvégezni, DB+1-től 0 — 5+1 pont

B. 1. megoldás: A legutolsó napra, amikor még elvégezhető — 5 pont
2. megoldás: A legelső napra, amikor elvégezhető (az első DB napra teszi a munkákat) — 5 pont

C. {*}: M*N; {**}: M^2, {***}: N — 3+3+3 pont

D. $N \leq M/2$ esetén kisebb az első algoritmus lépésszáma — 4 pont

2013. Második forduló

Ötödik-nyolcadik osztályosok

1. feladat: Vizsga (50 pont)

A bemenetet érdemes azonnal percre átszámolni, majd a kiírásnál a percet visszaalakítani (óra, perc) párra.

A leghosszabb 60 perces időszak valamely vizsgázó vizsgájának végétől kezdődik:

```
Leghosszabb(n,kezdet,vég,mego):
  k:=1; hossz:=-1; ii:=1
  Ciklus i=1-től n-1-ig
    ii:=max(i,ii)
    Ciklus amíg ii≤n és vég[ii]-vég[i]≤60
      ii:=ii+1
    Ciklus vége
    Ha ii-i+1>hossz akkor hossz:=ii-i+1; k:=i
  Ciklus vége
  mego:=vég[k]
Eljárás vége.
```

A második részfeladat egy egyszerű maximumkiválasztás

```
Vizsgaszünet(n,kezdet,vég,hossz):
  hossz:=0
  Ciklus i=1-től n-1-ig
    Ha kezdet[i+1]-vég[i]>hossz akkor hossz:=kezdet[i+1]-vég[i]
  Ciklus vége
Eljárás vége.
```

Szünetmentes időszak akkor kezdődik, amikor egy vizsga kezdete nagyobb, mint az előző vizsga vége. Ezen időszakok közül kell a leghosszabb.

```
Szünetmentes(n,kezdet,vég,hossz):
  hossz:=-1; k:=1
  Ciklus i=2-től n-ig
    Ha kezdet[i]≠vég[i-1]
       akkor Ha vég[i-1]-kezdet[k]>hossz
                akkor hossz:=vég[i-1]-kezdet[k]
             k:=i
  Ciklus vége
  Ha vég[n]-kezdet[k]>hossz akkor hossz:= vég[n]-kezdet[k]
Eljárás vége.
```

2. feladat: Számok (50 pont)

A feladat számintervallumokban levő számok lehetséges összegéről, különbségéről és szorzatáról szól. Minden számintervallum alsó és felső határának is ismerjük a számlálóját és a nevezőjét (a rekordok `asz`, `an`, `fsz`, `fn` nevű mezői).

```
Összead(x,y,a)
  a.asz:=x.asz*y.an+y.asz*x.an; a.an:=x.an*y.an
  a.fsz:=x.fsz*y.fn+y.fsz*x.fn; a.fn:=x.fn*y.fn
  Egyszerűsít(a)
Eljárás vége.
```

```
Kivon(x,y,b):
  b.asz:=x.asz*y.fn-y.fsz*x.an; b.an:=x.an*y.fn
  b.fsz:=x.fsz*y.an-y.asz*x.fn; b.fn:=x.fn*y.an
  Egyszerűsít(b)
Eljárás vége.
```

```
Szoroz(x,y,c):
  c.asz:=x.asz*y.asz; c.an:=x.an*y.an
  c.fsz:=x.fsz*y.fsz; c.fn:=x.fn*y.fn
  Egyszerűsít(c)
Eljárás vége.
```

```
Egyszerűsít(x):
  Ha x.asz=0 akkor x.an:=1
  különben i:=2
           Ciklus amíg i≤x.asz és i≤x.an
             Ha x.asz mod i=0 és x.an mod i=0
                akkor x.asz:=x.asz div i; x.an:=x.an div i
             különben i:=i+1
  Elágazás vége
  Ha x.fsz=0 akkor x.fn:=1
  különben i:=2
           Ciklus amíg i≤x.fsz és i≤x.fn
             Ha x.fsz mod i=0 és x.fn mod i=0
                akkor x.fsz:=x.fsz div i; x.fn:=x.fn div i
             különben i:=i+1
  Elágazás vége
Eljárás vége.
```

<u>3. feladat</u>: Autók (50 pont)

Ez egy szimulációs feladat, az autók mozgását kell követni! Az n+1. pozícióra (zebra utáni hely) érő autók a kilépők n időegység múlva. Az n. pozícióról akkor léphetnek tovább, ha a lámpa nem piros. Tovább lépni akkor lehet, ha nem ér hozzá az autó az előző autóhoz. Belépni szintén csak ilyen feltétellel lehet.

```
Autók(n,b,p,u,be,db,ki):
  ut[1..n+1]:=(0,…,0); db:=0; idő:=0; j:=1
  Ciklus amíg db<b
    idő:=idő+1
    Ha ut[n+1]=1 akkor db:=db+1; ki[db]:=idő+n; ut[n+1]:=0
    Ha ut[n]=1 akkor {nem piros lámpa}
       Ha (idő-1) mod p<p-u akkor ut[n+1]:=1; ut[n]:=0
    Ciklus i:=n-1-től 1-ig -1-esével
      Ha ut[i]=1 és ut[i+2]=0 akkor ut[i+1]:=1; ut[i]:=0
    Ciklus vége
    Ha idő≥be[j] és ut[1]+ut[2]=0 akkor ut[1]:=1; j:=j+1
  Ciklus vége
Eljárás vége.
```

Kilencedik-tizedik osztályosok

<u>1. feladat</u>: Iskola (40 pont)

Érdemes előfeldolgozást végezni, megadni minden napra az aznap órát tartó tanárok halmazát, továbbá minden tanárnak minden napra az első és utolsó órája sorszámát, valamint órái számát. Ugyancsak szükséges lehet az egyes tantárgyakat tanító tanárok halmaza, valamint egy tanár-, nap- és órasorszámmal indexelt mátrix, amiben tantárgy sorszámokat tárolunk.

```
Előfeldolgozás(n,tanár,tárgy,nap,óra):
  naphalmaz:=([],…,[]); tárgyhalmaz:=([],…,[])
  első[1..n,1..5]:=(9,…,9); utolsó[1..n,1..5]:=(0,…,0)
  Ciklus i=1-től n-ig
    naphalmaz[nap[i]]:=naphalmaz[nap[i]]∪[tanár[i]]
    db[tanár[i],nap[i]]:=db[tanár[i],nap[i]]+1
    Ha óra[i]<első[tanár[i],nap[i]]
                  akkor első[tanár[i],nap[i]]:=óra[i]
    Ha óra[i]>utolsó[tanár[i],nap[i]]
                  akkor utolsó[tanár[i],nap[i]]:=óra[i]
    órák[tanár[i],nap[i],óra[i]]:=tárgy[i]
    tárgyhalmaz[tárgy[i]]:=tárgyhalmaz[tárgy[i]]∪[tanár[i]]
  Ciklus vége
Eljárás vége.
```

Az első részfeladat megoldása a naphalmazok elemszáma. Ha a halmazokra van `elemszám` művelet, akkor a belső ciklus ezzel az egy művelettel megspórolható (a következő eljárásban is).

```
Órát tartók(n,naphalmaz,db):
  Ciklus i=1-től 5-ig
    db[i]:=0
    Ciklus j=1-től n-ig
      Ha j∈naphalmaz[i] akkor db[i]:=db[i]+1
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

A második feladat a legnagyobb elemszámú tárgyhalmaz elemszáma.

```
Legtöbb tanár(n,m,tárgyhalmaz,max):
  max:=0; maxért:=0
  Ciklus j=1-től m-ig
    d:=0
    Ciklus i=1-től n-ig
      Ha i∈tárgyhalmaz[j] akkor d:=d+1
    Ciklus vége
    Ha d>maxért akkor maxért:=d; max:=j
  Ciklus vége
Eljárás vége.
```

A harmadik feladatban meg kell számolni minden tanárra és minden napra a lyukasórák számát, majd ezekből kell egy maximumkiválasztás.

```
Lyukasórák(n,elso,utolso,db,sorsz):
  k:=0; maxért:=-1; sorsz:=-1
  Ciklus i=1-től n-ig
    d:=0
    Ciklus j=1-től 5-ig
      Ha db[i,j]>0 akkor d:=d+utolso[i,j]-elso[i,j]+1-db[i,j]
    Ciklus vége
    Ha d>maxért és d>0 akkor maxért:=d; sorsz:=i
  Ciklus vége
Eljárás vége.
```

A negyedik feladatban olyan tanárt kell keresni, akinek soha nincs a T tanárral együtt órája.

```
Helyettesítő(t,n,helyettes):
  i:=1; Ha i=t akkor i:=i+1
  Ciklus amíg i≤n és nem helyettesíthet(i,t)
    i:=i+1; Ha i=t akkor i:=i+1
  Ciklus vége
  Ha i≤n akkor helyettes:=I különben helyettes:=-1
Eljárás vége.
```

```
helyettesíthet(i,t):
  rossz:=hamis
  Ciklus j=1-től 5-ig
    Ciklus k=0-tól 8-ig
      Ha órák[t,j,k]>0 és órák[i,j,k]>0 akkor rossz:=igaz
    Ciklus vége
  Ciklus vége
  helyettesíthet:=nem rossz
Függvény vége
```

<u>2. feladat</u>: Újság (30 pont)

A folyóiratok terjesztése fastruktúrában történik, a fát a levelek felől ábrázoljuk. Az első feladat egy rekurzív függvénnyel oldható meg, memorizálással kerüljük el a többszörös kiszámolást (a `darab[a]` értéke -1, ha még nem volt kiszámolva).

```
hány(a,n,honnan):
  Ha darab[a]<0
     akkor s:=0
           Ciklus i=1-től n-ig
             Ha honnan[i]=a akkor s:=s+1+hány(i)
           Ciklus vége
           darab[a]:=s
  Elágazás vége
  hány:=darab[a]
Függvény vége
```

A második feladatban ki kell számolni az `a` csomópont távolságát a központi helytől.

```
átrakás(a,n,honnan,t):
  tav:=0; i:=a; t[a]:=0
  Ciklus amíg honnan[i]>0
    tav:=tav+1; i:=honnan[i]; t[i]:=tav
  Ciklus vége
  Ha tav=0 akkor átrakás:=tav különben átrakás:=tav-1
Eljárás vége.
```

A harmadik feladatban felhasználhatjuk a másodikban kiszámolt `t` vektort (feltéve, hogy az ott el nem ért elemekben 0 van). A `b` pontból elindulunk visszafelé, amíg olyan pontba nem érünk, amibe már az `a`-ból is mentünk.

```
Utolsó közös(a,b,honnan,t,közös):
  Ha t[b]>0 akkor közös:=b
  különben i:=b
           Ciklus amíg i≠a és honnan[i]>0 és t[i]=0
             i:=honnan[i]
           Ciklus vége
           közös:=i
  Elágazás vége
Eljárás vége.
```

3. feladat: Autó (35 pont)

Ez egy szimulációs feladat, az autók mozgását kell követni! Az `n+1`. pozícióra (zebra utáni hely) érő autók a kilépők `n` időegység múlva. Az `n-1`. pozícióról akkor léphetnek tovább, ha a kereszteződés szabad (jobbkézszabályt tartva). Tovább lépni akkor lehet, ha nem ér hozzá az autó az előző autóhoz. Belépni szintén csak ilyen feltétellel.

Ábrázolás:

`utv`, `utf` – a vízszintes és függőleges út állapota (1=autó, 0=üres)
`vbedb,fbedb` – a vízszintesen és függőlegesen belépések száma
`vbe,fbe` – a vízszintesen és függőlegesen belépések tervezett időpontja
`vkidb,fkidb` – a vízszintesen és függőlegesen kilépettek száma
`vki,fki` – a vízszintesen és függőlegesen kilépések időpontjai

```
  vi:=1; fi:=1; vkidb:=0; fkidb:=0
  idő:=0
  Ciklus amíg vkidb+fkidb<vbedb+fbedb
    idő:=idő+1
    kilépés(utv,kiv,vkidb); {kilépés jobbra}
    kilépés(utf,kif,fkidb); {kilépés lent}
  {kereszteződésbe lépések}
    utb[n+1]:=utb[n]; utb[n]:=0
    utf[n+1]:=utf[n]; utf[n]:=0
  {kereszteződés}
    Ha utb[n-1]=1 és utf[n+1]+utf[n]=0 {balról beléphet}
       akkor utb[n-1]:=0; utb[n]:=1
    utleptetes(utb,vbe,vbedb)
    Ha utf[n-1]=1 és utb[n+1]+utb[n]=0 {fentről beléphet}
       akkor utf[n-1]:=0; utf[n]:=1
    utleptetes(utf,fbe,fbedb)
  Ciklus vége
Eljárás vége.
```

Mindenki tovább lép, ha tud (nem érne hozzá az előtte levőhöz). Ha az idő ott tart, akkor belép az útra a következő (ha van helye).

```
utleptetes(ut,be,index,darab):
  Ciklus i=n-2-től 1-ig -1-esével
    Ha ut[i]=1 és ut[i+2]=0 akkor ut[i+1]:=1; ut[i]:=0
    Ha index≤darab és idő≥be[index])
       akkor Ha ut[1]+ut[2]=0 akkor ut[1]:=1; index:=index+1
  Ciklus vége
Eljárás vége.
```

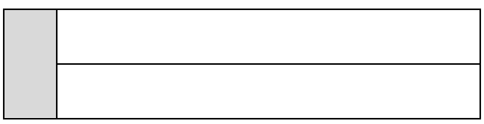
A kilépőket (út végén állók) kigyűjtjük, majd az út végét kiürítjük.

```
kilépés(ut,ki,index):
  Ha ut[n+1]=1 akkor index:=index+1; ki[index]:=idő+n
                     ut[n+1]:=0
Eljárás vége.
```

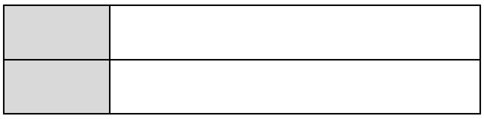
4. feladat: Járdakövezés (45 pont)

Klasszikus közvetett rekurziós feladat, kétféle típusú járda lefedését kell megoldani.

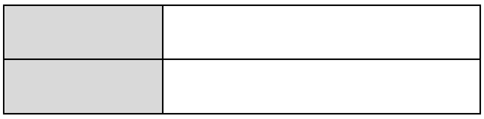
A lefedés indulhat így:



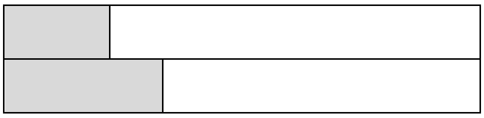
vagy így:



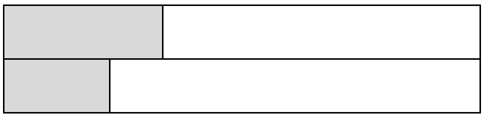
vagy így:



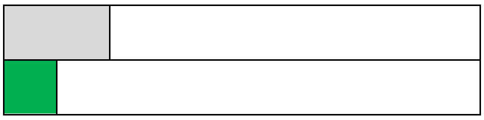
Lehet azonban egymás alá két különböző járdalapot tenni:



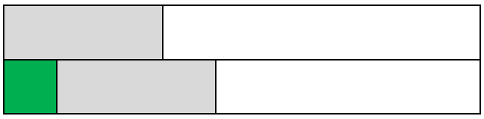
vagy így:



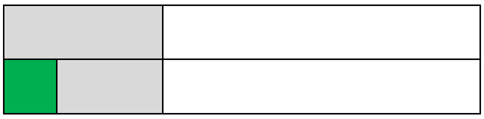
Az utóbbiak egy B-típusú járdát adnak maradékul, aminek az egyik bal sarka le van fedve. Ez a következőképpen folytatható:



vagy így:



végül így:



```
Járda(n):
  A[1]:=1; A[2]:=2; A[3]:=4
  B[1]:=0; B[2]:=0; B[3]:=1
  Ciklus i=4-től n-ig
    A[i]:=A[i-1]+A[i-2]+A[i-3]+2*B[i-2]
    B[i]:=A[i-3]+B[i-1]+B[i-3]
  Ciklus vége
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Tanár (32 pont)

Érdemes előfeldolgozást végezni, megadni minden napra az aznap órát tartó tanárok halmazát, továbbá minden tanárnak minden napra az első és utolsó órája sorszámát, valamint órái számát. Ugyancsak szükséges lehet az egyes tantárgyakat tanító tanárok halmaza, valamint egy tanár-, nap- és órasorszámmal indexelt mátrix, amiben tantárgy sorszámokat tárolunk.

```
Előfeldolgozás(n,tanár,tárgy,nap,óra):
  naphalmaz:=([],…,[]); tárgyhalmaz:=([],…,[])
  első[1..n,1..5]:=(9,…,9); utolsó[1..n,1..5]:=(0,…,0)
  Ciklus i=1-től n-ig
    naphalmaz[nap[i]]:=naphalmaz[nap[i]]∪[tanár[i]]
    db[tanár[i],nap[i]]:=db[tanár[i],nap[i]]+1
    Ha óra[i]<első[tanár[i],nap[i]]
                   akkor első[tanár[i],nap[i]]:=óra[i]
    Ha óra[i]>utolsó[tanár[i],nap[i]]
                   akkor utolsó[tanár[i],nap[i]]:=óra[i]
    órák[tanár[i],nap[i],óra[i]]:=tárgy[i]
    tárgyhalmaz[tárgy[i]]:=tárgyhalmaz[tárgy[i]]∪[tanár[i]]
  Ciklus vége
Eljárás vége.
```

Az első részfeladat megoldása a tanárszámból kivonva a naphalmazok elemszáma. Ha a halmazokra van `elemszám` művelet, akkor a belső ciklus ezzel az egy művelettel megspórolható (a következő eljárásban is).

```
Szabadnaposak(n,m,naphalmaz,d):
  Ciklus i=1-től 5-ig
    d[i]:=n
    Ciklus j=1-től n-ig
      Ha j∈naphalmaz[i]) akkor d[i]:=d[i]-1
    Ciklus vége
Eljárás vége.
```

A második részfeladatnál ki kell számolni minden tanár lyukasórái számát, majd ezek minimumát kell venni

```
Legkevesebb lyukasóra(n,első,utolsó,db,tmin):
  tmin:=0; min:=1000
  Ciklus i=1-től n-ig
    d:=0
    Ciklus j=1-től 5-ig
      Ha db[i,j]>0 akkor d:=d+utolso[I,j]-elso[I,j]+1-db[i,j]
    Ciklus vége
    Ha d<main akkor min:=d; tmin:=i
  Ciklus vége
Eljárás vége.
```

A harmadik részfeladatban olyan tanárt kell keresni, akinek a T tanárral egyszerre soha nincs órája. Először olyat, aki a T tanár egyik munkanapján sem szabadnapos. Ha nincs ilyen, akkor szabadnapos is jó.

```
Helyettesítő(t,n,orak,naphalmaz,h):
  i:=1; Ha i=t akkor i:=i+1
  Ciklus amíg i≤n és nem helyettesíthet(t,i)
    i:=i+1; Ha i=t akkor i:=i+1
  Ciklus vége
  Ha i≤n akkor h:=i
  különben i:=1; Ha i=t akkor i:=i+1
           Ciklus amíg i≤n és nem helyettesíthet2(t,i)
             i:=i+1; Ha i=t akkor i:=i+1
           Ciklus vége
             Ha i≤n akkor h:=i különben h:=-1
  Elágazás vége
Eljárás vége.
```

```
helyettesíthet(t,i):
  rossz:=hamis
  Ciklus j=1-től 5-ig
    Ha t∈naphalmaz[j] és i∉naphalmaz[j]) akkor rossz:=igaz
    különben Ciklus k=0-tól 8-ig
               Ha órák[t,j,k]>0 és órák[i,j,k]>0
                  akkor rossz:=igaz
             Ciklus vége
  Ciklus vége
  helyettesíthet:=nem rossz
Függvény vége.
```

```
helyettesíthet2(t,i):
  rossz:=hamis
  Ciklus j=1-től 5-ig
    Ciklus k=0-tól 8-ig
      Ha órák[t,j,k]>0 és órák[i,j,k]>0 akkor rossz:=igaz
    Ciklus vége
  Ciklus vége
  helyettesíthet2:=nem rossz
Függvény vége.
```

```
Szakos helyettesítés(t,h,n,órák,tárgyhalmaz,hedb,he):
  hedb:=0; j:=0
  Ciklus amíg j≤8 és hedb≠-1
    Ha órák[t,h,j]>0
        akkor i:=1; Ha i=t akkor i:=i+1
              Ciklus amíg i≤n és (órák[t,h,j]∉tárgyhalmaz[i])
                          vagy órák[i,h,j]>0)
                i:=i+1; Ha i=t akkor i:=i+1
              Ciklus vége
              Ha i≤n akkor hedb:=hedb+1; he[hedb]:=i
                 különben hedb:=-1
    j:=j+1
  Ciklus vége
Eljárás vége.
```

2. feladat: Modul (28 pont)

A feladat modellje egy irányított gráf, aminek A és B pontja (modulja) között akkor van él, ha az A modul közvetlenül használja a B modult. A pontokhoz (modulokhoz) szerzőket rendelünk.

Az első részfeladat az M pontból elérhető pontok megadása, pl. szélességi bejárással.

```
Fordítás(m,n,gráf,db,p):
  szín:=(fehér,…,fehér)
  Sorüres; Sorba(m); szín[m]:=szürke; db:=0
  Ciklus amíg nem üressor?
    Sorból(i)
    Ciklus j∈gráf[i]
      Ha szín[j]=fehér akkor Sorba(j); szín[j]:=szürke
                             db:=db+1; p[db]:=j
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

A második részfeladatban azon pontok szerzői halmazát kell megadni, ahonnan vezet közvetlen él az M pontba. (A második ciklus nem kell, ha a halmaz típusra van elemszám művelet.)

```
Felszólítás(m,n,gráf,szerző,szdb,sz):
  sz:=[]
  Ciklus j=1-től n-ig
    Ha m∈gráf[j] akkor sz:=sz∪[szerző[j]]
  Ciklus vége
  sz:=sz-[szerző[m]]
  szdb:=0
  Ciklus i=1-től m-ig
    Ha i∈sz akkor szdb:=szdb+1
  Ciklus vége
Eljárás vége.
```

Olyan szerző keresése, aki nem használ közvetlenül más által írt modult.

```
Keres(s,n,gráf,szer):
  szer:=1
  Ciklus amíg szer≤s és nem jó(szer)
    szer:=szer+1
  Ciklus vége
  Ha szer>s akkor szer:=-1
Eljárás vége.
```

```
jó(i):
  joo:=igaz
  Ciklus j=1-től n-ig
    Ha szerző[j]=i
       akkor Ciklus k∈gráf[j]
               Ha szerző[k]≠i akkor joo:=hamis
             Ciklus vége
  Ciklus vége
  jó:=joo
Függvény vége.
```

<u>3. feladat</u>: Zebra (30 pont)

Ez egy szimulációs feladat, egy mátrixban kell követni elemek mozgását. A mátrix sorai számát lefelé és felfelé is növeljük meg K-val és úgy helyezzük el oda az érkező gyalogosokat, hogy időegységenként lépve egyet, pontosan a kellő időben lépjenek be a zebrára! Egy elem értéke 1, ha az ott levő gyalogos lefelé tart, illetve 2, ha felfelé!

```
Zebra(n,m,k,z,ered):
  Ciklus idő=1-től n+k*2-ig
    db:=0
    Ciklus j=1-től m-ig {kilépők}
      Ha z[n+k,j]=1 akkor db:=db+1; z[n+k,j]:=0
      Ha z[k+1,j]=2 akkor db:=db+1; z[k+1,j]:=0
    Ciklus vége
    Ha idő>n akkor ered[i-n]:=db
```

```
  {felfelé}
    Ciklus i=n+k-1-től 1-ig -1-esével
      Ciklus j=m-től 1-ig -1-esével
        Ha z[i,j]=1 és z[i+1,j]=0 és z[i+2,j]≠2 {előre lép}
           akkor z[i,j]:=0; z[i+1,j]:=-1
        különben ha z[i,j]=1 és z[i+1,j]=0 és z[i+2,j]=2
                                              {felső nem lép}
                    akkor z[i,j]:=0; z[i+1,j]:=-1; z[i+2,j]:=3
      Ciklus vége
    Ciklus vége
    Ciklus i=2-től n+2*k-ig
      Ciklus j=1-től m-ig
        Ha z[i,j]=2 és z[i-1,j]=0 {felső lép}
           akkor z[i-1,j]:=3; z[i,j]:=0
      Ciklus vége
    Ciklus vége
    Ciklus i=n+k-tól 1-ig -1-esével
      Ciklus j=m-től 1-ig -1-esével
        Ha j<m és z[i,j]=1 és z[i+1,j]=2 és z[i,j+1]=0
                                              {jobbra kitér}
           akkor z[i,j+1]:=1; z[i,j]:=3; z[i+1,j]:=0
        különben ha z[i,j]=1 és z[i+1,j]=2
                    akkor z[i+1,j]:=4
        különben ha z[i,j]=-1 akkor z[i,j]:=1
      Ciklus vége
    Ciklus vége
    Ciklus i=2-től n+2*k-ig
      Ciklus j=1-től m-ig
        Ha z[i,j]=3 akkor z[i,j]:=2
        különben ha j>1 és z[i,j]=4 és z[i,j-1]=0 {balra kitér}
                    akkor z[i,j-1]:=2; z[i,j]:=1; z[i-1,j]:=0
        különben ha z[i,j]=4 akkor z[i,j]:=2
      Ciklus vége
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

4. feladat: Csoportkép (25 pont)

Ez egy klasszikus mohó stratégiás feladat (fényképezés) egy variációja, amiben darabszám korlát is van. Legyen `Int[u]` az u-ban távozó, legkésőbb érkező vendég érkezési időpontja!

```
Előfeldolgozás:
  Int:=(0,…,0)
  Ciklus i=1-től n-ig
    Ha e[i]>Int[u[i]] akkor Int[u[i]]:=e[i]
  Ciklus vége
Eljárás vége.
```

```
Fényképész(Int,D,T,k,):
  Utolsó:=0 {az utolsó fényképezési időpont}
  Első:=1 {az alkalom első f. időpontja}
  K:=0 {a megoldás elemszáma}
  hány:=D
  Ciklus x=1-től maxidő-ig
    Ha Int[x]>0 és Utolsó<Int[x]
       akkor ha hány=D akkor K:=K+1; Végére(M[K],x-1)
                             Első:=x-1; Utolsó:=x-1
                             hány:=1; ii:=ii+1
             különben ha x≤Első+T
                       akkor Utolsó:=x-1; Végére(M[K],Utolsó)
                             hány:=hány+1
             különben ha Int[x]<Első+T
                       akkor Utolsó:=Első+T-1
                             Végére(M[K],Utolsó); hány:=hány+1
             különben K:=K+1; Utolsó:=x-1; Első:=Utolsó
                      Végére(M[K],Utolsó); hány:=1
  Ciklus vége
Eljárás vége.
```

5. feladat: Számösszegzés (35 pont)

Legyen `E[x]` igaz, ha az x érték előállítható az a elemeiből!

```
  K:=a[n]
  E:=(hamis,…,hamis); E[0]=igaz; j:=1
  Ciklus x=1-től K-1-ig
    i:=1
    Ciklus amíg i≤n és nem (a[i]≤x és E[x-a[i]])
      i:=i+1
    Ciklus vége
    Ha i≤n akkor E[x]=igaz
    Ha x=b[j] akkor j:=j+1
                    Ha E[x] akkor ered[j]:=1
                         különben ered[j]:=0
  Ciklus vége
  Ciklus x=K-tól b[m]-ig
    i:=1
    Ciklus i=1-től n-ig
      xr:=(x-a[i]) mod K
    amíg i≤n és nem E[xr]
    Ciklus vége
    Exx:=i≤n
    Ha x=b[j] akkor j:=j+1
                    Ha Exx akkor ered[j]:=1
                        különben ered[j]:=0
    E[x mod K]=Exx
  Ciklus vége
Eljárás vége.
```

2013. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Ember (64 pont)

A feladat modellje egy fa, de egyik részfeladathoz sem kell ábrázolni a fát, elég a szülő-gyerek kapcsolatokat nyilvántartani.

Az első részfeladatban leszámoljuk mindenkire, hogy hány gyereke van, majd vesszük ezek maximumát.

```
Legtöbb gyerekes(n,m,szülő,max):
  gy:=(0,…,0)
  Ciklus i=1-től m-ig
    db[szülő[i]]:=db[szülő[i]]+1
  Ciklus vége
  max:=1
  Ciklus i=2-től n-ig
    Ha db[i]>db[max] akkor max:=i
  Ciklus vége
Eljárás vége.
```

A második részfeladatban mindenki minden gyerekére leszámoljuk, hogy hány gyereke van.

```
Legtöbb unokás(n,m,szülő,gyerek,max,u):
  u:=(0,…,0)
  Ciklus i=1-től m-ig
    Ciklus j=1-től m-ig
      Ha gyerek[i]=szülő[j] akkor u[szülő[i]]:=u[szülő[i]]+1
    Ciklus vége
  Ciklus vége
  max:=1
  Ciklus i=2-től n-ig
    Ha u[i]>u[max] akkor max:=i
  Ciklus vége
Eljárás vége.
```

A harmadik részfeladat a másodikban kiszámolt `u` vektor (kinek hány unokája van) alapján egy egyszerű megszámolás.

```
Unokátlan(n,u,db):
  db:=0
  Ciklus i=1-től m-ig
    Ha u[i]=0 akkor db:=db+1
  Ciklus vége
Eljárás vége.
```

2. feladat: Tördelés (56 pont)

Első lépésként ki kell hagyni a {}-jelek közötti részeket, kivéve a {{}} közöttieket, a dupla zárójelekből egyet kell hagyni! Az első betű is elhagyandó.

```
Előkészítés(s,t):
  t:=''; i:=2
  Ciklus amíg i≤hossz(s)-1
    Ha s[i]+s[i+1]='{{' vagy s[i]+s[i+1]='}}'
       akkor t:=t+s[i]; i:=i+2
    különben ha s[i]='{' akkor Ciklus amíg s[i]≠'}'
                                 i:=i+1
                               Ciklus vége
    különben t:=t+s[i]; i:=i+1
  Ciklus vége
Eljárás vége.
```

Az átalakított szöveg első K sorát L szóközzel kell kezdeni! A sort a sorba kiférő utolsó szó után törjük meg.

```
Nyomtatás(t,k,l,m):
  db:=0; j:=1
  Ciklus j≤hossz(t)
    Ha db>k akkor i:=j+m különben i:=m+j-L
    Ciklus amíg i>hossz(t)
      i:=i-1
    Ciklus vége
    Ciklus amíg i>0 és t[i]≠' '
      i:=i-1
    Ciklus vége
    db:=db+1
    Ha db>k akkor sor[db]:=t[j..i-1]
         különben sor[db]:=szóközök[L]+t[j..i-1]
    j:=i+1
    Ciklus amíg j≤hossz(t) és t[j]=' '
      j:=j+1
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

3. feladat: Út (30 pont)

A megoldásban egy N elemű logikai tömböt használunk, az igaz értékűek nem voltak felújítva.

```
Aszfaltozás(n,m,k,v):
  u:=(igaz,…,igaz)
  Ciklus i=1-től m-ig
    Ciklus j=k[i]-től v[i]-1-ig
      u[j]:=hamis
    Ciklus vége
  Ciklus vége
  h:=0
  Ciklus i=0-tól n-1-ig
    Ha u[i] akkor h:=h+1
  Ciklus vége
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Épület (20 pont)

A megoldás a teljes külső falból indul ki, majd levonja a baloldali, a felső, a jobboldali és az alsó beépített falakat.

```
Épület(n,fx,fy,ax,ay.bfx,bfy,jax,jay,h,max):
  h:=2*(ax-fx+1)+2*(ay-fy+1)
  alsó:=(hamis,…,hamis); felső:=(hamis,…,hamis)
  bal:=(hamis,…,hamis); jobb:=(hamis,…,hamis)
  Ciklus i=1-től n-ig
    Ha bfx[i]=fx akkor h:=h-(jay[i]-bfy[i]+1)
                       bal[bfy[i]..jay[i]]:=igaz
    Ha jax[i]=ax akkor h:=h-(jay[i]-bfy[i]+1)
                       jobb[bfy[i]..jay[i]]:=igaz
    Ha bfy[i]=fy akkor h:=h-(jax[i]-bfx[i]+1)
                       felső[bfx[i]..jax[i]]:=igaz
    Ha jay[i]=ay akkor h:=h-(jax[i]-bfx[i]+1)
                       alsó[bfx[i]..jax[i]]:=igaz
  Ciklus vége
  max[1]:=Szakasz(bal,fy,ay)
  max[2]:=Szakasz(alsó,fx,ax)
  max[3]:=Szakasz(jobb,fy,ay)
  max[4]:=Szakasz(felső,fx,ax)
Eljárás vége.
```

```
Szakasz(t,k,v):
  Ha t[1] akkor db:=0 különben db:=1
  Ciklus i=k+1-től v-ig
    Ha t[i-1] és nem t[i] akkor db:=db+1
  Ciklus vége
  Szakasz:=db
Függvény vége.
```

2. feladat: Robot (32 pont)

Egy fagráfot építünk, ahol minden csomóponthoz megadjuk, hogy megy-e ki belőle olyan csatorna, amelyen a robot elfér, illetve megy-e bele. Robot onnan indulhat, ahova nem megy be ebben a gráfban él és megy ki belőle.

```
Robot(n,cs,cspszám,db):
  be:=(hamis,…, hamis); ki:=(hamis,…, hamis); hova:=(0,…,0)
  Ciklus i=1-től n-1-ig
    hova[cs[i,1]]:=cs[i,2]
    Ha cs[i,3]>r akkor ki[cs[i,1]]:=igaz; be[cs[i,2]]:=igaz
  Ciklus vége
  cspszám:=0
  Ciklus i=1-től n-ig
    Ha nem be[i] és ki[i] akkor cspszám:=cspszám+1
  Ciklus vége
  Ciklus i=1-től n-ig
    Ha nem be[i] és ki[i] akkor db[i]:=0; j:=i
                                Ciklus amíg hova[j]>0
                                  db[i]:=db[i]+1; j:=hova[j]
                                Ciklus vége
  Ciklus vége
Eljárás vége.
```

3. feladat: Hírlánc (32 pont)

A 0 befokú pontok biztos részei az eredménynek, de így kimaradnának azon körök, ahova nem vezet be út 0 befokú pontból. A feladat tehát egy olyan ponthalmaz meghatározása (dominátor halmaz), amelyből az összes pont elérhető. Ehhez kétszer járjuk be a gráfot. Az első bejárás kezdőpontjaiból ellenkező sorrendű bejárásokat indítva a második bejárások kezdőpontjai lesznek a dominátor halmaz elemei.

```
Hírlánc(n,a,b):
  Szin:=(fehér,…,fehér); Dn:=0
  Ciklus i=1-től n-ig
    Ha Szin[i]=fehér akkor Bejár(i,fekete); Dn:=Dn+1; D[Dn]:=i
  Ciklus vége
  M:=0
  Ciklus i=Dn-től 1-ig -1-esével
    Ha Szin[D[i]]=fekete akkor m:=m+1; Mego[m]:=D[i]
                               Bejár(D[i],piros)
  Ciklus vége
Eljárás vége.
```

```
Bejár(p,Sz)
  Szin[p]:=Sz
  Ha Szin[A[p]]≠Sz akkor Bejár(A[p],Sz)
  Ha Szin[B[p]]≠Sz akkor Bejár(B[p],Sz)
Eljárás vége.
```

4. feladat: Számok (32 pont)

Rendezzük sorba a lefedendő számokat! Az i. szám akkor tartozik egy megkezdett intervallumba, ha az előtt levő üres szakasz hossza kisebb, mint az őt követőé.

```
Számok(n,P,db,mego):
  M=(-1,…,-1); Rendezés(n,P)
  fedő=0; P[n+1]=2*P[n]; e=1; i=3
  Ciklus amíg i≤n+1
    Ha i=n-1 akkor fedő:=fedő+P[i-1]-P[e]; M[e]=i-1
                   fedő:=fedő+P[n]-P[i]; M[i]=n
    különben Ciklus amíg i<n és P[i]-P[i-1]<P[i+1]-P[i])
                i:=i+1
             Ciklus vége
             fedő:=fedő+P[i-1]-P[e]; M[e]=i-1; e=i; i+=2
  Ciklus vége
  i=1; db:=0;
  Ciklus amíg i≤n
    ii=M[i]; db:=db+1; mego[db,1]:=P[i]; mego[db,2]:=P[ii]
    i=ii+1
  Ciklus vége
Eljárás vége.
```

5. feladat: Gépek (34 pont)

Ez egy dinamikus programozásos feladat, amiben minden alkatrésznél tároljuk, hogy mennyi volt az optimális idő, ha az előzőt az első, illetve a második gépen gyártottuk. Ebből kiszámolhatjuk az optimumot az aktuális alkatrész után. Az optimumokból így az aktuálisra és az azt megelőzőre van csak szükség.

```
Gépek(K,N,AB,BA,A,B,x):
  Opt[igaz,1..K]:=(0,…,0)
  régi:=hamis
  Ciklus i=1-től N-ig
    új:=régi; régi:=nem új
    Ciklus j=1-től K-ig
      Opt[új,1]:=Opt[régi,1]+A[x[i]]
      Ha Opt[régi,2]+BA+A[x[i]]<Opt[új,1]
         akkor Opt[új,1]:=Opt[régi,2]+BA+A[x[i]]
      Opt[új,2]:=Opt[régi,2]+B[x[i]]
      Ha Opt[régi,1]+AB+B[x[i]]<Opt[új,2]
         akkor Opt[új,2]:=Opt[régi,1]+AB+B[x[i]]
    Ciklus vége
  Ciklus vége
  Ha Opt[új,1]<Opt[új,2] akkor ered:=Opt[új,1]
                      különben ered:=Opt[új,2]
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Ház (20 pont)

Tegyünk a t mátrixban a szobák helyére igaz értéket! Az új szobáknak biztosan van bal felső sarka, amitől jobbra és lefelé meg kell keresni a következő szobát (tudjuk, hogy a szabad területek is biztosan téglalap alakúak).

```
Ház(n,ax,ay,fx,fy):
  t[fy..ay,fx..ax]:=hamis
  t[fy..ay,fx-1]:=igaz; t[fy-1,fx..ax]:=igaz
  Ciklus k=1-től n-ig
    t[bfy[k]..jay[k],bfx[k]..jax[k]]:=igaz
  Ciklus vége
  db:=0; ter:=0
  Ciklus i=fy-tól ay-ig
    Ciklus j=fx-től ax-ig
      Ha nem t[i,j]
         akkor Ha t[i-1,j] és t[i,j-1]
                  akkor db:=db+1
                        k:=i+1
                        Ciklus amíg k≤ay és nem t[k,j]
                          k:=k+1
                        Ciklus vége
                        l:=j+1
                        Ciklus amíg l≤ax és nem t[i,l]
                          l:=l+1
                        Ciklus vége
                        Ha ter<(k-i)*(l-j)
                          akkor ter:=(k-i)*(l-j)
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

2. feladat: Csatorna (30 pont)

Egy gráfot építünk a csatornahálózatból, de csak azokkal az élekkel, amelyekbe a robot befér. Az első részfeladat az S pontot tartalmazó komponens elemszáma, a második pedig a gráf komponenseinek száma.

```
Csatorna(n,s,gráf,fok,sk,ok):
  szín:=(fehér,…,fehér)
  Bejár(s,sk)
  ok:=0
  Ciklus i=1-től n-ig
    Ha szín[i]=fehér Bejár(i,a)
                     Ha a>0 akkor ok:=ok+1
  Ciklus vége
Eljárás vége.
```

A `Bejár` eljárás bármelyik gráfbejárás lehet, ami a bejárás során a második paraméterében visszaadja a komponens elemszámát.

3. feladat: Poligon (30 pont)

A konvex burok azon oldalai két végpontját kell megadni, amelyen nincs a törtvonalnak másik pontja. A konvex burok előállításában kihasználhatjuk, hogy a pontok már rendezve vannak vagy az óramutató járásával megegyező, vagy azzal ellentétes irányban. Az előbbi esetén érdemes a pontok sorrendjét megfordítani.

```
Poligon(N,P,db,ered):
  k:=sarokpont(N,P)
  Ha k>1 akkor előző:=k-1 különben előző:=n
  Ha k<n akkor következő:=k+1 különben következő:=1
  Ha Fordul(előző,k,következő)=1 akkor Sorrendfordít(n,P,k)
  P[n+1]:=P[1]; Körbejár(N,P,k,bdb,b); b[db+1]:=b[1]
  Ciklus i=1-től bdb-ig
    Ha jó(i) akkor db:=db+1; ered[db]:=(b[i],b[i+1])
  Ciklus vége
Eljárás vége.

Sarokpont(N,P):
  s:=1
  Ciklus i=2-től N-ig
    Ha P[i].x<P[s].x vagy P[i].x=P[s].x és P[i].y<P[s].y
       akkor s:=i
  Ciklus vége
  Sarokpont:=s
Függvény vége.
```

```
Körbejár(N,P,k,db,B):
  Ha k<n akkor köv:=k+1 különben köv:=1
  Ha köv<n akkor köv2:=köv+1 különben köv2:=1
  Ciklus amíg Fordul[P[1],P[köv],P[köv2]]=0
    köv:=köv2
    Ha köv<n akkor köv2:=köv+1 különben köv2:=1
  Ciklus vége
  B[1]:=k; B[2]:=köv; db:=2
  Ciklus amíg köv≠k
    Ha Fordul(P[B[db-1]],P[B[db]],P[köv])≥0 akkor db:=db-1
       különben db:=db+1; B[db]:=köv
    Ha köv<n akkor köv:=köv+1 különben köv:=1
  Ciklus vége
  db:=db-1
Eljárás vége.
```

```
jó(i):
  j:=b[i]; k:=j+1
  Ciklus amíg k≠b[i+1] és Fordul(P[j],P[k],P[b[i+1]])≠0
    k:=k+1
  Ciklus vége
  jó:=(k=b[i+1])
Függvény vége.
```

4. feladat: Lefedés (30 pont)

Rendezzük sorba a lefedendő számokat! Keressük meg a rendezett sorozatban a szomszédos elemek K-1 legnagyobb különbségét! Ezzel a számhalmazt K részre bontottuk, megadtuk a K intervallumot, amelyek összhossza a lehető legkisebb. Ha valamelyik intervallum 1 elemet tartalmazna, a hossza akkor is 1 kell legyen!

```
Lefedés(k,n,x,h,kezd,veg):
  Ciklus i=1-től n-ig
    P[i].x=x; P[i].vége:=hamis
  Ciklus vége
  Rendez(n,P)
  Ciklus i=1-től n-1-ig
    táv[i].az:=i; táv[i].tav=P[i+1].x-P[i].x
  Ciklus vége
  Rendez(n-1,tav)
  Ciklus i=1-től k-1-ig
    P[R[i].az].vége=igaz
  Ciklus vége
  j:=1; kezd[j]:=P[1].x; h:=0
  Ciklus i=1-től n-1-ig
    Ha p[i].vége akkor vég[j]:=P[i].x
                       Ha kezd[j]=vég[j] akkor h:=h+1
                       különben h:=h+vég[j]-kezd[j]
                       j:=j+1, kezd[j]:=P[i+1].x
  Ciklus vége
  vég[j]:=P[n].x
  Ha kezd[j]=vég[j] akkor h:=h+1
                 különben h:=h+vég[j]-kezd[j]
Eljárás vége
```

5. feladat: Játék (40 pont)

Ez egy dinamikus programozásos feladat a játéktáblát leíró gráfon. Ki kell számolni minden legfeljebb K lépésszámos játékra, minden p mezőre, hogy onnan indulva mekkora lehet Ádám maximális nyeresége!

```
Játék(k,n,t,g,nyereség):
  Opt[régi,1..n]:=(0,…,0) , régi:=igaz; új:=hamis
  Ciklus i=1-től K-ig
    Ciklus p=1-től n-ig
      Opt[új,p]=-∞
      Ciklus q∈G[p]
        min=∞
        Ciklus r∈G[q]
          Ha Opt[régi,r]-t[r]<min akkor min=Opt[régi,r]-t[r]
        Ciklus vége
        Ha Opt[új,p]<min+t[q]) akkor Opt[új,p]=min+t[q]
      Ciklus vége
    Ciklus vége
    régi=új; új=nem régi
  Ciklus vége
  nyereség:=Opt[régi,1]
Eljárás vége.
```

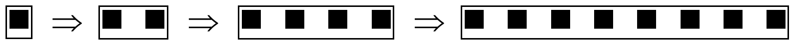
2014. Első forduló

Ötödik-nyolcadik osztályosok

Számítógép nélküli feladatok

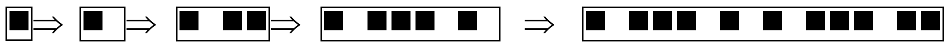
1. feladat: Sorminta (40 pont)

A. 1 ⇒ 101 ⇒ 1010101 ⇒ 101010101010101 — 3+4+5 pont



B. 1 ⇒ 10 ⇒ 1001 ⇒ 10010110 ⇒ 1001011001101001 — 2+3+4+5 pont

D. 1⇒ 10⇒ 1011⇒ 10111010 ⇒ 1011101010111011 — 2+3+4+5 pont



2. feladat: Mit csinál (40 pont)

A. X=2	5 pont
B. X=20	5 pont
C. X=4	5 pont
D. X=10	5 pont
E. X a kapott A, B, C érték közül a középső lesz	10 pont
F. Abban az esetben hamis a második feltétel, ha C nem kisebb sem A-nál, sem B-nél (C a legnagyobb válasz esetén 5 pont adható)	10 pont

3. feladat: Számok (60 pont)

A. A=5,B=4	5+5 pont
B. A=5, B=5	5+5 pont
C. A a legnagyobb érték, B a második legnagyobb (lehet A-val egyforma)	5+5 pont
D. A=5, B=3	5+5 pont
E. B az A-nál kisebbek közül a legnagyobb (azaz A-val nem lehet egyforma) (5 pont levonás ebből, ha A-ra is ad változást)	10 pont
F. Bármely bemenet jó, amelyben az X vektor elemei egyformák	10 pont

Számítógépes feladat – VÁLASZTHATÓ

4. feladat: Települések (60 pont)

Az első részfeladat egy maximumkiválasztás:

```
Kukutyintól(n,kuktáv,s):
  s:=1
  Ciklus i=2-től n-ig
    Ha kuktáv[i]>kuktáv[s] akkor s:=i
  Ciklus vége
Eljárás vége.
```

A második feladat egy megszámolás:

```
Közelebb(n,kuktáv,pirtáv,db):
  db:=0
  Ciklus i=1-től n-ig
    Ha kuktáv[i]>pirtáv[i] akkor db:=db+1
  Ciklus vége
Eljárás vége.
```

A harmadik feladat egy kiválogatás:

```
Kiválogat(n,kuktáv,pirtáv,db,sor):
  db:=0
  Ciklus i=1-től n-ig
    Ha pirtáv[i]<100 és kuktáv[i]≥100
       akkor db:=db+1; sor[db]:=i
  Ciklus vége
Eljárás vége.
```

Számítógép nélküli feladat – VÁLASZTHATÓ

<u>4. feladat</u>: Települések (60 pont)

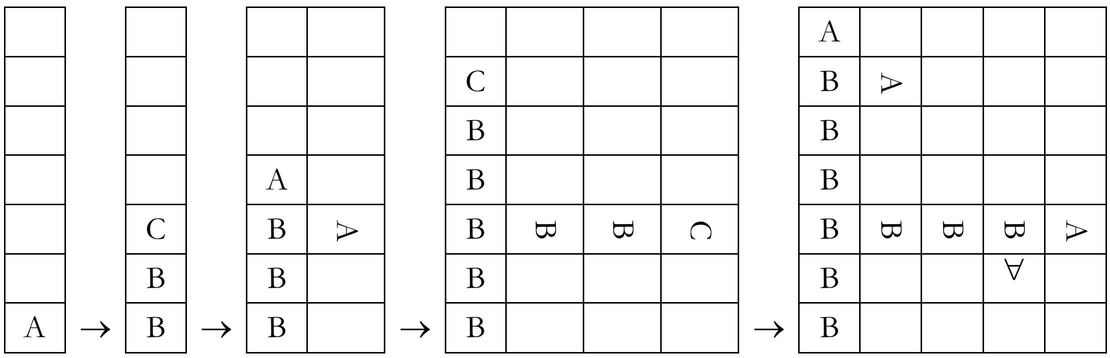
```
Település(N,K,P,A,B,C,L):
  A:=0; B:=1; D:=0; L:=0
  Ciklus i=1-től N-ig
    Ha P(i)≤100 akkor A:=A+P(i); D:=D+1             6+6+6 pont
    Ha K(i)+P(i)<K(B)+P(B) akkor B:=i                 6+6 pont
    Ha P(i)<K(i) akkor L:=L+1; C(L):=P(i)           6+6+6 pont
  Ciklus vége
  Ha D>0 akkor A:=A/D különben A:=0                   6+6 pont
Eljárás vége.
```

Kilencedik-tizedik osztályosok

<u>1. feladat</u>: Növény (42 pont)

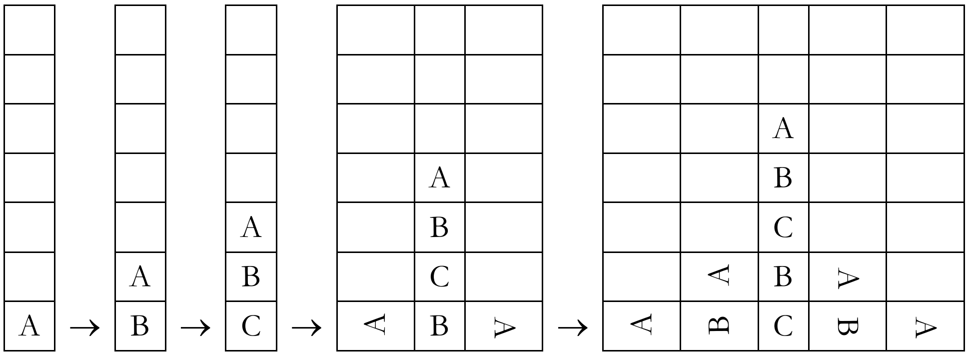
A. 18+8 pont

(ahány betűt helyesen helyettesít a következő ábrában, illetve ahány teljes ábra jó*2, azaz pl. az 1. időegységre az egy A betűt kellett helyettesíteni és ha jó, akkor az 1. időegység ábrája is jó, tehát erre a részre 1+2 pont jár; a 2. időegységre a BBC betűket kellett helyettesíteni, ami 3 pont lehet, további 2 pont, ha jó a 2. időegység ábrája, azaz mind a három helyettesítés jó volt)



B. 12 +4 pont

(ahány betűt helyesen helyettesít a következő ábrában, illetve ahány teljes ábra jó)



<u>2. feladat</u>: Mit csinál (51 pont)

A. D=1, kivonások száma: 6 6+7 pont

B. D=4, kivonások száma: 3 6+7 pont

C. D=6, kivonások száma: 6 6+7 pont

D. Az eredmény a három bemeneti érték legnagyobb közös osztója lesz 12 pont

<u>3. feladat</u>: Adatok (67 pont)

A. A=3, B=13, C=3, E=(2,1,4) 5+5+5+5 pont

B. A az első 0 értékű elem indexe, B az utolsó 0 értékű elem indexe, C a 0-kat tartalmazó szakaszok száma, E(i) értékei pedig az ilyen szakaszok hossza 8+8+8+8 pont

C. Ha az X vektorban nincs 0 5 pont

D. Ha az X vektorban egyetlen 0 van 5 pont

E. Ha az X vektor minden második eleme 0, minden második pedig nem 0 5 pont

<u>4. feladat</u>: Békák (40 pont)

A lentiektől különböző más jó lépéssorrend is lehet:

A. lépésszám: 3, a lépések: B lép, Z ugrik, B lép VAGY Z lép, B ugrik, Z lép 5+10 pont
(ha hosszabb, de jó megoldást ad, akkor 0+5 pont adható)

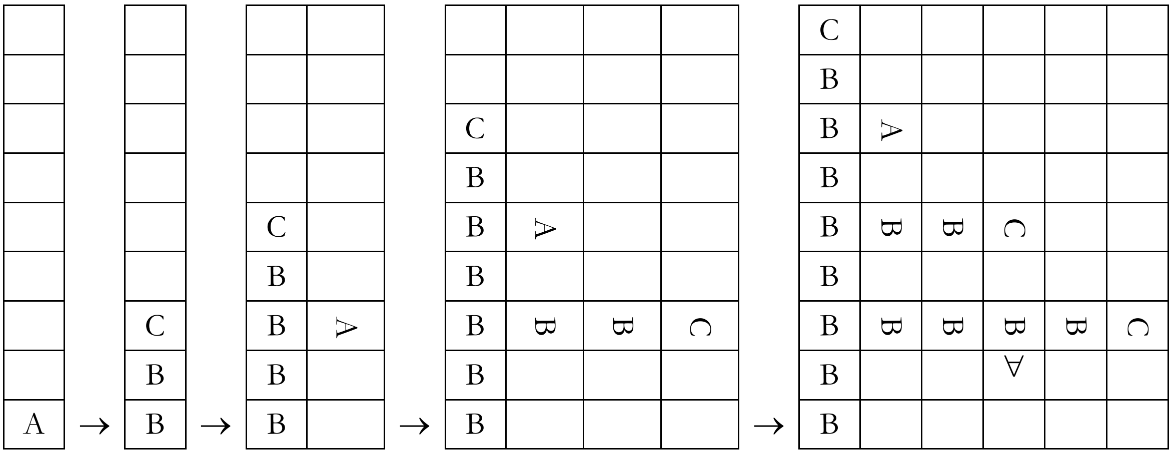
B. lépésszám: 8, a lépések: B lép, Z ugrik, Z lép, B ugrik, B ugrik, Z lép, Z ugrik, B lép (vagy ugyanez Z és B felcserélésével) 10+15 pont
(ha hosszabb, de jó megoldást ad, akkor 0+10 pont adható)

Tizenegyedik-tizenharmadik osztályosok

<u>1. feladat</u>: Növény (37 pont)

A. 21 pont

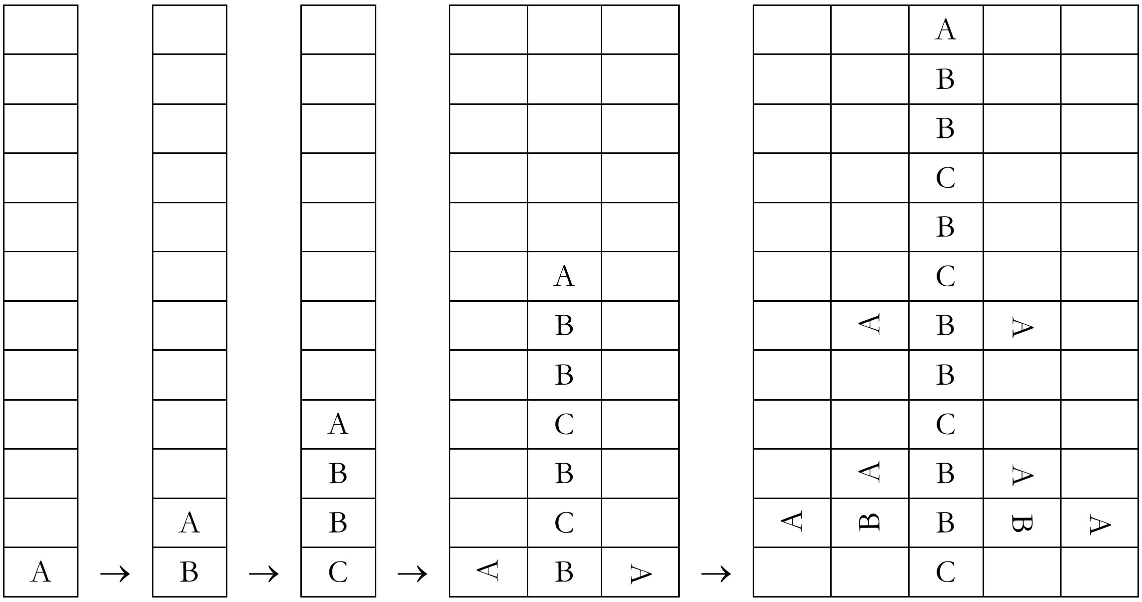
(ahány betűt helyesen helyettesít a következő ábrában – azaz pl. az 1. időegységre az egy A betűt jól helyettesíti, akkor erre a részre 1 pont jár; a 2. időegységre a BBC betűket kellett helyettesíteni, ami 3 pont lehet, ha mind a három helyettesítés jó volt)



A → BBC → BBB(A)BC → BBB(BBC)BB(A)BC → BBB(BBB(A)BC)BB(BBC)BB(A)BC

B. 16 pont

(ahány betűt helyesen helyettesít a következő ábrában)



A → BA → CBBA → B[A](A)CBCBBA → CB[BA](BA)B[A](A)CBB[A](A)CBCBBA

<u>2. feladat</u>: Mit csinál (60 pont)

A. D=30, összeadások száma: 28 — 6+6 pont

B. D=16, összeadások száma: 4 — 6+6 pont

C. D=252, összeadások száma: 38 — 6+6 pont

D. Az eredmény a három bemeneti érték legkisebb közös többszöröse lesz — 12 pont

E. Az összeadások száma=D/A+D/B+D/C-3 — 12 pont

3. feladat: Adatok (48 pont)

A. A=3, B=(2,3,3), C=(4,3,5) 5+5+5 pont

B. A a nem 0-kat tartalmazó szakaszok száma, B(i) értékei az ilyen szakaszok hosszai (elemszámai), C(i) értékei pedig az ilyen szakaszok maximumértékei 5+5+5 pont

C. Ha az X vektorban minden elem 0 5 pont

D. D az aktuális, nem 0-kból álló szakasz hossza; E az aktuális, nem 0-kból álló szakasz maximuma 4+4 pont

E. Ha az X vektor minden második eleme 0, minden második pedig nem 0 5 pont

4. feladat: Békák (25 pont)

A lentiektől különböző más jó lépéssorrend is lehet:

A. lépésszám: 5, a lépések egy lehetséges megoldásban: B lép, Z ugrik, B lép, B ugrik, Z lép 5+5 pont
(ha hosszabb, de jó megoldást ad, akkor 0+5 pont adható)

B. lépésszám: 8, a lépések: B lép, Z ugrik, Z lép, B ugrik, B ugrik, Z lép, Z ugrik, B lép (vagy ugyanez Z és B felcserélésével) 5+10 pont
(ha hosszabb, de jó megoldást ad, akkor 0+10 pont adható)

5. feladat: Geometria (30 pont)

Alfa: Db=4; Q=(2,3,6,8) 1+4 pont
Q: azon pontok sorszáma, amelyek az előzőhöz képest balra látszanak az elsőből 5 pont

Béta: Db=3; Q=(2,3,6) 1+4 pont
Q: azon pontok sorszáma, amelyek a másodikhoz képest balra látszanak az elsőből, és nem takarja őket szakasz 5 pont

Gamma: Db=6; Q=(2,3,4,5,6,7) 1+4 pont
Q: azon pontok sorszáma, amelyek a másodikhoz képest balra vagy jobbra látszanak az elsőből, és nem takarja őket szakasz 5 pont

2014. Második forduló

Ötödik-nyolcadik osztályosok

1. feladat: Mértékegységek (40 pont)

Az összeadást többjegyű helyiértékes számok összeadásaként végezzük:

```
Összead(a,b,c):
  Ha a.hüvelyk+b.hüvelyk<12
     akkor c.hüvelyk:=a.hüvelyk+b.hüvelyk; at:=0
  különben c.hüvelyk:=a.hüvelyk+b.hüvelyk-12; at:=1
  Ha a.láb+b.láb+at<3 akkor c.láb:=a.láb+b.láb+at; at:=0
                 különben c.láb:=a.láb+b.láb+at-3; at:=1
  Ha a.yard+b.yard+at<1760
     akkor c.yard:=a.yard+b.yard+at; at:=0
  különben c.yard:=a.yard+b.yard+at-1760; at:=1
  c.mérföld:=a.mérföld+b.mérföld+at
Eljárás vége.
```

A kivonáshoz érdemes megírni az angol mértékegységekre a NAGYOBB függvényt!

```
nagyobb(a,b):
  Ha a.mérföld>b.mérföld akkor nagyobb:=igaz
  különben ha a.mérföld=b.mérföld
     akkor Ha a.yard>b.yard akkor nagyobb:=igaz
           különben ha a.yard=b.yard
              akkor Ha a.láb>b.láb akkor nagyobb:=igaz
                    különben ha a.láb=b.láb és
                             a.hüvelyk>b.hüvelyk
                      akkor nagyobb:=igaz
                      különben nagyobb:=hamis
           különben nagyobb:=hamis
     különben nagyobb:=hamis
Függvény vége.
```

```
Kivon(a,b,c,jel):
  jel:=''
  Ha nagyobb(b,a) akkor Csere(a,b); jel:='-'
  Ha a.hüvelyk-b.hüvelyk≥0
     akkor d.hüvelyk:=a.hüvelyk-b.hüvelyk; at:=0
  különben d.hüvelyk:=a.hüvelyk-b.hüvelyk+12; at:=1
  Ha a.láb-b.láb-at≥0 akkor d.láb:=a.láb-b.láb-at; at:=0
                 különben d.láb:=a.láb-b.láb-at+3; at:=1
  Ha a.yard-b.yard-at≥0
     akkor d.yard:=a.yard-b.yard-at; at:=0
  különben d.yard:=a.yard-b.yard-at+1760; at:=1
  d.mérföld:=a.mérföld-b.mérföld-at
Eljárás vége.
```

2. feladat: Kínai Nagy Fal (55 pont)

Meg kell számolni azon szomszédos őrhelyek számát, amelyek mindkét végén van őrség (mindkét szám 1-es), amelyek egyik végén van őrség (az egyik szám 1-es), valamint a leghosszabb szakaszt, ami két védett hely között van.

```
Fal(n,t,v,o,lh):
  v:=0; o:=0; u:=1; lh:=0
  Ciklus i=1-től n-1-ig
    Ha t[i]*t[i+1]=1 akkor v:=v+1
    Ha t[i]+t[i+1]>0 akkor o:=o+1
    Ha t[i]*t[i+1]=1 akkor Ha i-u>lh akkor lh:=i-u
                           u:=i+1
  Ciklus vége
  Ha n-u>lh akkor lh:=n-u
Eljárás vége.
```

3. feladat: Játék (55 pont)

Észrevehető, hogy bármely sor `j`-edik eleme alatt a következő sor `2*j-1`-edik és `2*j`-edik eleme van, így minden lépés után elég tárolni az aktuális sor- és oszlopindexet.

```
Játék(k,lépés,s,o):
  s:=1; o:=1; i:=1; hiba:=hamis
  Ciklus amíg i≤k és nem hiba
    Ha lépés[i]='BL' akkor s:=s+1; o:=o*2-1
    különben ha lépés[i]='JL' akkor s:=s+1; o:=o*2
    különben ha lépés[i]='F'
                akkor ha s=1 akkor hiba:=igaz
                          különben s:=s-1; o:=(o+1) div 2
    különben ha lépés[i]='B'
                akkor ha o=1 akkor hiba:=igaz
                                   különben o:=o-1
    különben ha lépés[i]='J'
                akkor ha o=hatvány(s-1) akkor hiba:=igaz
                                        különben o:=o+1
    i:=i+1
  Ciklus vége
Eljárás vége.
```

```
hatvány(s):
  h:=1
  Ciklus i=1-től s-ig
    h:=h*2
  Ciklus vége
  hatvány:=h
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Fal (44 pont)

A `t` tömb 0 ás 1 értékű elemeket tartalmaz, 1 jelöli, hogy ott őrség található. Védett szakaszból annyi van, ahány védett szakasz kezdődik (egy 0 után két 1-es). Őrzött szakaszból annyi van, ahány őrzött szakasz kezdődik (két 0 után egy 1-es). A harmadik részfeladatban meg kell szüntetni az egymás melletti 0-kat!

```
Fal(n,t,v,o,ko):
  v:=0; o:=0; ko:=0; u:=0; őrzött:=hamis; t[0]:=0; t[n+1]:=0
  Ciklus i=0-tól n-1-ig
    Ha t[i]=0 és t[i+1]=1 és t[i+2]=1 akkor v:=v+1
    Ha nem őrzött akkor Ha t[i]=1 akkor o:=o+1; őrzött:=igaz
    különben Ha t[i]=0 és t[i+1]=0 akkor őrzött:=hamis
    Ha i>0 és t[i]=0 és t[i+1]=0
       akkor Ha u<i akkor ko:=ko+1; u:=i+1
  Ciklus vége
  Ha nem őrzött és t[n]=1 akkor o:=o+1
Eljárás vége.
```

2. feladat: Utcák (30 pont)

A feladat modellje egy gráf, amit ábrázolás helyett számítunk. Legyen `gráf[i,j].észak` igaz értékű, ha az `(i,j)` pontból lehet észak felé haladni! Hasonlóan legyenek `gráf[i,j].kelet`, `gráf[i,j].dél`, `gráf[i,j].nyugat` mezők is! Ezek a bemenet alapján kitölthetők. A kezdeti kitöltésben oldhatjuk meg azt is, hogy ne lehessen elhagyni a térképet – fiktív behajtani tilos táblák. Ezután nincs más tennivaló, mint egy szélességi bejárás erre a gráfra. Legyen `volt[i,j]` igaz, ha már jártunk az `(i,j)` pontban!

```
Utcák(gráf,ins,ino,ves,veo):
  s:=ins; o:=ino; Sorüres
  Sorba(s,o); volt[s,o]:=igaz
  Ciklus amíg nem üressor? és nem volt[ves,veo]
    Sorból(s,o)
    Ha gráf[s,o,észak] és nem volt[s+1,o]
       akkor Sorba(s+1,o); Honnan(s+1,o):=(s,o,észak)
             volt[s+1,o]:=igaz
    Ha gráf[s,o,kelet] és nem volt[s,o+1]
       akkor Sorba(s,o+1); Honnan(s,o+1):=(s,o1,kelet)
             volt[s,o+1]:=igaz
    Ha gráf[s,o,dél] és nem volt[s-1,o]
       akkor Sorba(s-1,o); Honnan(s-1,o):=(s,o,dél)
             volt[s-1,o]:=igaz
    Ha gráf[s,o,nyugat] és nem volt[s,o-1]
       akkor Sorba(s,o-1); Honnan(s,o-1):=(s,o,nyugat)
             volt[s,o-1]:=igaz
  Ciklus vége
  Bejár(ves,veo)
Eljárás vége.
```

Megjegyzés: A `Sorba(s,o)` a `volt[s,o]`-t igazra állítja.

```
Bejár(s,o,db):
 Ha s=ins and o=ino akkor Ki: db
 különben Bejár(honnan[s,o].s,honnan[s,o].o,db+1)
          ir:=honnan[s,o].ir
          Ha ir=észak akkor Ki: ”E”
          különben ha ir=dél akkor Ki: ”D”
          különben ha ir=kelet akkor Ki: ”K”
          különben akkor Ki: ”N”
Eljárás vége.
```

3. feladat: Játéktábla (40 pont)

Észrevehető, hogy a `j`-edik sorba `j` lépéssel le lehet jutni. Felfelé lépés azt jelenti, hogy felejtsük el az utolsó lefelé lépést! A balra lépés azt jelenti, hogy az utolsó jobbra le irányú lépést balra le irányúra változtatjuk, az úton levő közbülsőket pedig jobbra le irányúvá. Hasonlóan járunk el a jobbra lépésnél is.

```
Tábla(k,lépés,j,t):
  j:=0
  Ciklus i=1-től n-ig
    Ha s[i]<2 akkor j:=j+1; t[j]:=s
    különben ha s[i]=2 akkor j:=j-1
    különben ha s[i]=3 akkor balra(j)
    különben ha s[i]=4 akkor jobbra(j)
  Ciklus vége
Eljárás vége.
```

```
balra(j):
  k:=j
  Ciklus amíg t[k]=0
    t[k]:=1; k:=k-1
  Ciklus vége
  t[k]:=0
Eljárás vége.
```

```
jobbra(j):
  k:=j
  Ciklus amíg t[k]=1
    t[k]:=0; k:=k-1
  Ciklus vége
  t[k]:=1
Eljárás vége.
```

4. feladat: Mozi (36 pont)

Rendezzük sorba az igényeket ülőhely sorszám szerint (megőrizve a régi igénylő sorszámokat)! Ezután nézzük végig az ülőhelyeket! Átlépjük azokat az igénylőket, akiknek ez már túl nagy lenne. Ha a következő igénylőnek ez a hely túl kicsi, akkor vesszük a következő ülőhelyet, különben odaadjuk neki.

```
Mozi(n,s,m,k,db,er):
  Rendez(n,s)
  x:=1; db:=0; i:=1
  Ciklus amíg i≤m és x≤n
    Ciklus amíg x≤n és s[x]+k<i
      x:=x+1
    Ciklus vége
    Ha x≤n akkor Ha s[x]≤i akkor er[i]:=x; db:=db+1; x:=x+1
  Ciklus vége
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Nagy Fal (38 pont)

A t tömb 0 és 1 értékű elemeket tartalmaz, 1 jelöli, hogy ott őrség található.

```
Előkészítés(m,be,fel,t):
  fel:=0
  Ciklus i=1-től m-ig
    Ha t[be[i]]=1 akkor fel:=fel+1 különben t[be[i]]:=1
  Ciklus vége
Eljárás vége.
```

Védett szakaszból annyi van, ahány védett szakasz kezdődik (egy 0 után két 1-es). Őrzött szakaszból annyi van, ahány őrzött szakasz kezdődik (két 0 után egy 1-es).

```
Fal(n,t,v,o,ko):
  v:=0; o:=0; ko:=0; u:=0; őrzött:=hamis; t[0]:=0; t[n+1]:=0
  Ciklus i=0-tól n-1-ig
    Ha t[i]=0 és t[i+1]=1 és t[i+2]=1 akkor v:=v+1
    Ha nem őrzött akkor Ha t[i]=1 akkor o:=o+1; őrzött:=igaz
    különben Ha t[i]=0 és t[i+1]=0 akkor őrzött:=hamis
  Ciklus vége
  Ha nem őrzött és t[n]=1 akkor o:=o+1
Eljárás vége
```

A védetté tehető szakaszokhoz számoljuk meg, hogy milyen hosszú 0-kból álló szakaszból hány van! Könnyen belátható, hogy célszerű az egyetlen 0-t tartalmazó szakaszokkal kezdeni, mert ide 1-est elhelyezve két szakaszt teszünk védetté. Utána ugyanezen elv szerint jöhetnek a két 0-t tartalmazók, és így tovább. Az elején, illetve a végén levő 0 szakaszok a legrosszabbak, itt annyi szakasz tehető védetté, ahány 0-ból állnak.

```
Védetté(n,t,fel,vt):
  db:=0; vt:=0; e:=1; u:=0;
  Ciklus amíg e≤n és t[e]=0
    e:=e+1
  Ciklus vége
  Ha e≤n akkor Ciklus i=e-től n-ig
                 Ha t[i]=1 akkor x[db]:=x[db]+1; db:=0
                        különben db:=db+1
               Ciklus vége
               u:=db
  Elágazás vége
  i:=1; j:=fel
  Ciklus amíg j>0 és i≤n
    Ha j>i*x[i] akkor j:=j-i*x[i]; vt:=vt+(i+1)*x[i]
    különben vt:=vt+(i+1)*(j div i)+j mod i; j:=0
    i:=i+1
  Ciklus vége
  Ha j>0 akkor Ha j>e+u-1 akkor vt:=vt+e+u-1
                       különben vt:=vt+j
Eljárás vége.
```

Őrzötté tételnél először az egymás melletti három hosszú 0-kkal kell foglalkozni, a középsőt lecserélve két újabb falat tettünk őrzötté. Utána jöhetnek az egymás melletti kettő hosszú 0-k.

```
Őrzötté(n,t,fel,ot):
  ot:=0
  Ciklus i=1-től n-2-ig
    Ha t[i]+t[i+1]+t[i+2]=0 és fel>0
       akkor ot:=ot+2; t[i+1]:=1; fel:=fel-1
  Ciklus vége
  Ciklus i=1-től n-1-ig
    Ha t[i]+t[i+1]=0 és fel>0
       akkor ot:=ot+1; t[i+1]:=1; fel:=fel-1
  Ciklus vége
Eljárás vége.
```

2. feladat: Város (26 pont)

A feladat modellje egy gráf, amit ábrázolás helyett számítunk. Legyen `gráf[i,j,észak]` azon irányok halmaza, ahova lehet lépni, ha az `(i,j)` pontba északi irányba haladva érkeztünk. Hasonlóan legyenek `gráf[i,j,kelet]`, `gráf[i,j,dél]`, `gráf[i,j,nyugat]` is! Ezek a bemenet alapján kitölthetők. A kezdeti kitöltésben oldhatjuk meg azt is, hogy ne lehessen elhagyni a térképet – fiktív behajtani tilos táblák. Ezután nincs más tennivaló, mint egy szélességi bejárás erre a gráfra. Legyen `volt[i,j,irány]` igaz, ha már jártunk az `(i,j)` pontban `irány` irányból érkezve! (Most egy ponton többször is áthaladhatunk.

```
Utcák(n,m,gráf,ins,ino,ves,veo):
  s:=ins; o:=ino; Sorüres
  Ha s<n akkor Sorba(s+1,o,észak); volt[s+1,o,észak]:=igaz
  Ha s>1 akkor Sorba(s-1,o,dél); volt[s-1,o,dél]:=igaz
  Ha o<m akkor Sorba(s,o+1,kelet); volt[s,o+1,kelet]:=igaz
  Ha o>1 akkor Sorba(s,o-1,nyugat); volt[s,o-1,nyugat]:=igaz
  Ciklus amíg nem üressor? és nem
        (volt[ves,veo,észak] vagy volt[ves,veo,dél] vagy
         volt[ves,veo,kelet] vagy volt[ves,veo,nyugat])
    Sorból(s,o,irány)
    Ha észak∈gráf[s,o,irány]
       akkor Sorba(s+1,o,észak)
             Honnan(s+1,o,észak):=(s,o,irány)
             volt[s+1,o,észak]:=igaz
    Ha dél∈gráf[s,o,irány]
       akkor Sorba(s-1,o,dél)
             Honnan(s-1,o,dél):=(s,o,irány)
             volt[s-1,o,dél]:=igaz
    Ha kelet∈gráf[s,o,irány]
       akkor Sorba(s,o+1,kelet)
             Honnan(s,o+1,kelet):=(s,o,irány)
             volt[s,o+1,kelet]:=igaz
    Ha nyugat∈gráf[s,o,irány]
       akkor Sorba(s,o-1,nyugat)
             Honnan(s,o-1,nyugat):=(s,o,irány)
             volt[s,o-1,nyugat]:=igaz
  Ciklus vége
  Bejár(ves,veo)
Eljárás vége.
```

```
Bejár(s,o,db):
 Ha s=ins and o=ino akkor Ki: db
 különben Bejár(honnan[s,o,irány].s,honnan[s,o,irány].o,db+1)
          ir:=honnan[s,o,irány].ir
          Ha ir=észak akkor Ki: ”E”
          különben ha ir=dél akkor Ki: ”D”
          különben ha ir=kelet akkor Ki: ”K”
          különben akkor Ki: ”N”
Eljárás vége.
```

<u>3. feladat</u>: Játéktábla (26 pont)

Észrevehető, hogy a `j`-edik sorba `j` lépéssel le lehet jutni. Felfelé lépés azt jelenti, hogy felejtsük el az utolsó lefelé lépést! A balra lépés azt jelenti, hogy az utolsó középre le vagy jobbra le irányú lépést eggyel balrább le irányúra változtatjuk, az úton levő közbülsőket pedig jobbra le irányúvá. Hasonlóan járunk el a jobbra lépésnél is.

```
Tábla(k,lépés,j,t):
  j:=0
  Ciklus i=1-től n-ig
    Ha s[i]<3 akkor j:=j+1; t[j]:=s
    különben ha s[i]=3 akkor j:=j-1
    különben ha s[i]=4 akkor balra(j)
    különben ha s[i]=5 akkor jobbra(j)
  Ciklus vége
Eljárás vége.
```

```
balra(j):
  k:=j
  Ciklus amíg t[k]=0
    t[k]:=1; k:=k-1
  Ciklus vége
  t[k]:=t[k]-1
Eljárás vége.
```

```
jobbra(j):
  k:=j
  Ciklus amíg t[k]=2
    t[k]:=0; k:=k-1
  Ciklus vége
  t[k]:=t[k]+1
Eljárás vége.
```

<u>4. feladat</u>: Gépek (25 pont)

Elsőként számoljuk ki, hogy mely munkanapon hány határidős munkája van, majd ebből azt, hogy az adott napig hány határidős munka van! Mindre kiszámolhatjuk, hogy addig hány gépre lenne szükség, ezek maximuma a szükséges gépek száma. Ez alapján minden gépet folytonosan beosztunk.

```
Gépek(n,m,h,db):
  db:=(0,…,0)
  Ciklus i=1-től m-ig
    db[h[i]]:=db[h[i]]+1
  Ciklus vége
  tol[1]:=1
  Ciklus i=2-től n-ig
     tol[i]:=tol[i-1]+db[i-1]
  Ciklus vége
```

```
  db:=0; sum:=0
  Ciklus i=1-től n-ig
    sum:=sum+db[i]
    Ha (sum-1) div i+1>db akkor db:=(sum-1) div i+1
  Ciklus vége
  Ciklus i=1-től m-ig
    mego[i,1]:=(tol[h[i]]-1) div s+1
    mego[i,2]:=(tol[h[i]]-1) mod s+1
    tol[h[i]]:=tol[h[i]]+1
  Ciklus vége
Eljárás vége.
```

5. feladat: Fazekas (35 pont)

Számítsuk ki a `db[i]`-ben, hogy az `i`-edik terméket a két kemencében összesen hány termékkel együtt kell égetni az optimális megoldáshoz! A két kemencében egyszerre `2*k` termék lehet, azaz legfeljebb ennyi egyszerre kemencében levő termékre kell kiszámolnunk a legjobb esetet! Legyen `menet[i]` az első `i` tárgy optimális égetési ideje!

```
Fazekas(n,k,eg,idő):
  menet[0]:=0; menet[1]:=maxégetés
  menet[2]:=eg[1]+eg[2]; db[2]:=2
  Ciklus i=3-tól n-ig
    menet[i]:=eg[i]+eg[i-1]+menet[i-2]; db[i]:=2
    Ciklus j=3-tól 2*k-ig
      Ha i-j≥0 akkor Ciklus l=i-től i-j+1-ig -1-esével
                       r[i-l+1]:=eg[l]
                     Ciklus vége
                     Rendez(j,r)
                     Ha j<k+2 akkor
                        Ha menet[i]>r[j]+r[1]+menet[i-j]
                           akkor menet[i]:=r[j]+r[1]+menet[i-j]
                                 db[i]:=j
                     különben ha menet[i]>r[j]+r[j-k]+menet[i-j]
                         akkor menet[i]:=r[j]+r[j-k]+menet[i-j]
                               db[i]:=j
    Ciklus vége
  Ciklus vége
  idő:=menet[n]
Eljárás vége.

kiir(n,m):
  Ha n>1 akkor kiir(n-db[n],m);
               Ciklus i=n-db[n]+1-től n-ig
                 sor[i]:=i
               Ciklus vége
               Rendez(n-db+1,n,sor) {eg[sor[i]] szerint}
               Ha db[n]>k+1 akkor x:=n-k különben x:=n-db[n]+1
               Ciklus i=n-db[n]+1-től n-ig
                 Ha sor[i]≤x akkor Ki: m,1
                          különben Ki: m,2
               Ciklus vége
               m:=m+1
  különben m:=1
Eljárás vége.
```

2014. Harmadik forduló

Ötödik-nyolcadik osztályosok

1. feladat: Hofstadter (40 pont)

Az r vektorra van képletünk, az s vektorban kell követnünk a szabályt.

```
Hofstadter(n,r,s):
  r[1]:=1; s[1]:=2;r[2]:=3; s[2]:=4; u:=3
  Ciklus i=3-tól n-ig
    r[i]:=r[i-1]+s[i-1]; s[i]:=s[i-1]+1
    Ha s[i]=r[u] akkor s[i]:=s[i]+1; u:=u+1
  Ciklus vége
Eljárás vége.
```

2. feladat: Lottó (40 pont)

Számon kell tartanunk, hogy hányféle szám volt már az eddigi sorozatban, és ha mind a 90 volt, akkor az eredményhez hozzáadunk egyet, majd kezdjük elölről az előfordulások figyelését.

```
Lottó(n,s):
  volt:=(hamis,...,hamis); s:=0; v:=0
  Ciklus i=1-től n-ig
    Ciklus j=1-től 5-ig
      Ha nem volt[k[i,j]] akkor v:=v+1; volt[[i,j]]:=igaz
    Ciklus vége
    Ha v=90 akkor s:=s+1; v:=0; volt:=(hamis,...,hamis)

  Ciklus vége
Eljárás vége.
```

3. feladat: Sorrend (70 pont)

Kell keresni két azonosan induló szakaszt! A rövidebbhez illesszünk egyet a másikból (ez egyértelmű, mert mindkét szalagon különböző számok voltak), és így tovább!

```
Sorrend(n,m,első,masod):
  eh:=0; mh:=0; i:=1; j:=1
  Ciklus amíg első[i].elem[1]≠masod[j].elem[1]
    ha j<m akkor j:=j+1 különben i:=i+1; j:=1
  Ciklus vége
  eh:=eh+első[i].db; mh:=mh+masod[j].db
  Ha első[i].db>masod[j].db akkor Másol(első[i],1,elso[i].db)
                              különben Másol(masod[j],1,masod[j].db)
  Ciklus amíg eh≠mh
    Ha eh<mh akkor k:=Kiválaszt(jo.elem[eh+1],első)
                   Ha eh+elso[k].db>mh
                      akkor Másol(első[k],mh-eh+1,első[k].db)
                   eh:=eh+elso[k].db
    különben k:=Kiválaszt(jo.elem[mh+1],masod)
             Ha mh+masod[k].db>eh
                akkor Másol(masod[k],eh-mh+1,masod[k].db)
             mh:=mh+masod[k].db
  Ciklus vége
Eljárás vége.
```

```
Kiválaszt(x,t):
  i:=1
  Ciklus amíg x≠t[i].elem[1]
    i:=i+1
  Ciklus vége
  Kiválaszt:=i
Függvény vége.
```

```
Másol(t,tol,ig):
  Ciklus i=tol-tól ig-ig
    jo.db:=jo.db+1; jo.elem[jo.db]:=t.elem[i]
  Ciklus vége
Eljárás vége.
```

Kilencedik-tizedik osztályosok

1. feladat: Metró (22 pont)

Ez egy egyszerű szimulációs feladat, amiben több változót kell követnünk időegységenként.

Jelölések:

`fv`	fent várakozók száma
`lv`	lent várakozók száma
`s`	szerelvények száma
`idő[x]`	az x. percben érkezők száma
`el[y]`	az y. szerelvénnyel elmenők száma

```
Metró(n,k,l,m,u,s,el):
  lépcső:=(0,…,0); idő:=(0,…,0)
  Ciklus i=1-től u-ig
    idő[s[i]]:=idő[s[i]]+1
  Ciklus vége
  fv:=0; lv:=0; s:=0; akt:=k-1; i:=1
  Ciklus amíg i≤n+k+m és lv≤l
    fv:=fv+idő[i]
    Ha i mod m=0                {jön a metró}
       akkor s:=s+1; el[s]:=lv; lv:=0
    lv:=lv+lépcső[akt];         {lépcsőről a váróterembe}
    Ha fv>2 akkor lépcső[akt]:=2 különben lépcső[akt]:=fv
    fv:=fv-lépcső[akt]          {fent várakozók a lépcsőre}
    akt:=(akt+1) mod k;         {lépcső mozgása}
    i:=i+1
  Ciklus vége
Eljárás vége.
```

2. feladat: Párosítás (22 pont)

Rendezzük mindkét csapatot erősség szerint csökkenő sorrendben! Ha a te csapatod aktuális tagja erősebb az ellenfél aktuális tagjánál, akkor összepárosítjuk őket. Ha nem tudja legyőzni, akkor a csapatodból senki sem tudja legyőzni. Akkor jársz legjobban, ha ilyenkor csapatod leggyengébb tagját párosítod vele.

```
Párosítás(n,a,b,pár):
  Rendez(n,a); Rendez(n,b); mego:=0; első:=1, utolsó:=n
  Ciklus i=1-től N-ig
    Ha a[első].erő≤b[i].erő akkor Pár[a[utolsó].s]=b[i].s
                                  utolsó:=utolsó-1
    különben mego:=mego+1; Pár[A[első].s]=B[i].s; első:=első+1
  Ciklus vége
Eljárás vége.
```

3. feladat: Mingrel ábécé (32 pont)

A mingrel szavak ábécésorrendje egy gráfot definiál a mingrel ábécé betűire. Ha egy szó a benne levő X betű miatt van előbb, mint az ugyanazon a helyen Y betűt tartalmazó másik szó, akkor az ábécében X megelőzi Y-t, azaz a gráfba vegyünk fel egy X→Y irányított élt! Ha a gráf elkészült, akkor a feladat ennek a gráfnak a topologikus rendezése.

```
Mingrel(n,szó,ábécé):
  befok:=(0,…,0); gráf:=(hamis,…,hamis)
  Ciklus i=1-től n-1-ig
    j:=1
    Ciklus amíg j≤hossz(szó[i]) és j≤hossz(szó[i+1]))
                és szó[i][j]=szó[i+1][j]
      j:=j+1
    Ciklus vége
    Ha j≤hossz(szó[i]) és j≤hossz(szó[i+1]))
        akkor gráf[szó[i][j],szó[i+1][j]]:=igaz
              befok[szó[i+1][j]]:=befok[szó[i+1][j]]+1
  Ciklus vége
  Sorüres; db:=0
  Ciklus c:='a'-tól 'z'-ig
    Ha befok[c]=0 akkor Sorba(c)
  Ciklus vége
  Ciklus amíg nem üressor?
    Sorból(c); db:=db+1; ábécé[db]:=c
    Ciklus d='a'-tól 'z'-ig
      Ha gráf[c,d] akkor befok[d]:=befok[d]-1
                         Ha befok[d]=0 akkor Sorba(d)
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

4. feladat: Dobozok (32 pont)

Állítsunk elő egy irányított gráfot, amelyben akkor van él az `i.` pontból a `j.` pontba, ha az `i.` dobozba betehető a `j.` doboz! Az egyszerűsítés kedvéért a dobozok három méret paraméterét rendezzük csökkenő sorrendbe!

```
Gráf létrehozása(n,d,g,befok):
  befok:=(0,…,0); g:=(üres,…,üres)
  Ciklus i=1-től n-ig
    Ciklus j=1-től n-ig
      Ha d[i].x>d[j].x és d[i].y>d[j].y és d[i].z>d[j].z)
         akkor befok[j]:=befok[j]+1; g[i]:=g[i]∪{j}
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Ezután a feladat megtalálni ebben a gráfban a leghosszabb utat. A leghosszabb út biztosan olyan pontból indul, amibe nem lehet belerakni másik dobozt, azaz a gráfban az ilyen pontnak nincs bemenő éle. A gyors kiszámítás érdekében érdemes előállítani egy topologikus rendezést.

```
Toprend(n,g,hely):
  Sorüres; t:=0
  Ciklus i=1-től n-ig
    Ha befok[i]=0 akkor Sorba(i); t:=t+1; hely[t]:=i
  Ciklus vége
  Ciklus amíg nem üressor?
    Sorból(p)
    Ciklus j∈g[p]
      Befok[j]:=Befok[j]-1
      Ha Befok[j]=0 akkor Sorba(j); t:=t+1; hely[t]:=j
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Rekurzió memorizálással: számítsuk ki egy tetszőleges pontból a kivezető leghosszabb utat!

```
Úthossz(i):
  Ha tav[i]=0
     akkor ha üres?(g[i]) akkor tav[i]:=1; kov[i]:=0
  különben h:=0; p:=g[i];
           Ciklus p∈g[i]
             Úthossz(p)
             ha tav[p]>h akkor h:=tav[p]; kov[i]:=p
           Ciklus vége
           tav[i]:=h+1
Eljárás vége.
```

Végül következik a leghosszabb út megtalálása. Mivel a memorizálás miatt minden pontra legfeljebb egyszer számítjuk ki a belőle kivezető leghosszabb utat, ezért a pontokat itt tetszőleges sorrendben vizsgálhatjuk.

```
Dobozok(n,d):
  tav:=(0,…,0)
  Gráf létrehozása(n,d,g,befok)
  Toprend(n,g,hely)
  h:=0
  Ciklus i=1-től n-ig
    Úthossz(hely[i])
    Ha tav[hely[i]]>h akkor h:=tav[hely[i]]; v:=hely[i]
  Ciklus vége
  Ki: h
  Ciklus amíg v>0
    Ki: v; v:=kov[v]
  Ciklus vége
Eljárás vége.
```

5. feladat: Szállítás (42 pont)

Indítsunk egy topologikus rendezést a fa azon pontjaiból, ahova nem megy be él! Számítsuk ki közben minden pontra, hogy oda hány kamion érkezik be

```
Szállítás(n,hova,befok,termel,tárol,kamionszám,méret):
  kamion:=(0,…,0); Sorüres
  Ciklus i=1-től n-ig
    Ha befok[i]=0 akkor Sorba(i);
  Ciklus vége
  Ciklus amíg nem üressor?
    Sorból(i)
    Ha termel[i]>tarol[i]
        akkor kk:=termel[i]-tarol[i]
              Ha kamion[i]<(kk-1) div k+1 és hova[i]>0
                 akkor kamion[i]:=(kk-1) div k+1
    különben kk:=0
    kamion[hova[i]]:=kamion[hova[i]]+kamion[i]
    termel[hova[i]]:=termel[hova[i]]+kk
    befok[hova[i]]:=befok[hova[i]]-1
    Ha befok[hova[i]]=0 és hova[i]>0 akkor Sorba(hova[i])
  Ciklus vége
  kamionszám:=kamion[i]); méret:=termel[i]-tarol[i]
Eljárás vége.
```

Tizenegyedik-tizenharmadik osztályosok

1. feladat: Metró (20 pont)

A feladat az előző korcsoportos hasonló feladat nehezítése, itt a felfelé menő mozgólépcsőt is kezelni kell, valamint a felfelé menő lépcsőre várakozókat is.

Jelölések:

`fv`	fent várakozók száma
`lv`	lent várakozók száma
`lentv`	lent a lépcsőre várakozók száma
`s`	szerelvények száma
`idő[x]`	az x. percben érkezők száma
`el[y]`	az y. szerelvénnyel elmenők száma
`le[y]`	az y. szerelvényről leszállók száma

```
Metró(n,k,l,m,u,s)
  lépcső:=(0,...,0) ; idő:=(0,...,0)
  Ciklus i=1-től u-ig
    idő[s[i]]:=idő[s[i]]+1
  Ciklus vége
  fv:=0; lv:=0; s:=0; akt:=k-1; lentv:=0; i:=1
  Ciklus amíg i≤n+k+m és lv+lentv≤l
    fv:=fv+idő[i]
    ha lentv>2 akkor lentv:=lentv-2 különben lentv:=0
    Ha i mod m=0                {jön a metró}
       akkor s:=s+1; el[s]:=lv; lv:=0
    lentv:=lentv+le[s]          {leszállók a metróról}
    lv:=lv+lépcső[akt];         {lépcsőről a váróterembe}
    Ha fv>2 akkor lépcső[akt]:=2 különben lépcső[akt]:=fv
    fv:=fv-lépcső[akt]          {fent várakozók a lépcsőre}
    akt:=(akt+1) mod k;         {lépcső mozgása}
    i:=i+1
  Ciklus vége
Eljárás vége.
```

<u>2. feladat</u>: Fénykép (30 pont)

Az érkezési idő szerinti sorrendben érkező vendégeket tegyük be egy távozási idő szerinti prioritási sorba! Ha a sor elemszáma legalább K, akkor az első K elemet kivehetjük. Az egyszerűsítés kedvéért vegyünk fel egy n+1-edik vendéget, aki az utolsó távozási időpontban érkezik! A feldolgozást a lehetséges időpontok szerint végezzük.

```
Fénykép(n,e,t,f,x):
  Prsorüres; f:=0; e[n+1]:=t[1]
  Ciklus i=2-től n-ig
    Ha t[i]>e[n+1] akkor e[n+1]:=t[i]
  Ciklus vége
  i:=1
  Ciklus idő=e[1]-től e[n+1]-ig
    Ciklus amíg nem üresprsor? és t[PrsorElső]<idő)
      Prsorból(j)
    Ciklus vége            {az idő előtt elmenők kiléptek}
    Ciklus amíg i≤n és e[i]=idő
      PrSorba(i); i:=i+1;  {az i-edik be a sorba}
    Ciklus vége
    Ha PrsorElemszám≥k akkor  {legalább K elem van a sorban}
       akkor f:=f+1
             Ciklus j=1-től k-ig
               Prsorból(x[f,j])
             Ciklus vége
  Ciklus vége
Eljárás vége.
```

<u>3. feladat</u>: Koncert (30 pont)

Rendezzük a bemenetet az első igényelt ülőhely sorszáma szerinti sorrendbe! Számítsuk ki dinamikus programozással, hogy az első i ülőhelyből az első j igény esetén maximum mennyit lehet kielégíteni!

```
Koncert(m,n,h,k,ered,db,t):
  u[1..m,0]:=(0,…,0); u[0,1..n]:=(0,…,0)
  Ciklus i=1-től m-ig
    Ciklus j=1-től n-ig
      u[i,j]:=max(u[i-1,j],u[i,j-1])
      Ha i≤h[j] és i≥k[j]
         akkor u[i,j]:=max(u[i,j],u[i-k[j],j-1]+k[j])
    Ciklus vége
  Ciklus vége
  i:=m; j:=n; db:=0
  Ciklus amíg i>0 és j>0
    Ciklus amíg i>0 és u[i,j]=u[i-1,j]
      i:=i-1
    Ciklus vége
    Ciklus amíg j>0 és u[i,j]=u[i,j-1]
      j:=j-1
    Ciklus vége
    db:=db+1; t[db,1]:=hely[j]; t[db,2]:= i-k[j]+1
    i:=i-k[j]; j:=j-1
  Ciklus vége
  ered:=u[m,n]
Eljárás vége.
```

<u>4. feladat</u>: Robot (30 pont)

Számítsuk ki, hogy az i. raktárig eljutva mi a legjobb megoldás, majd vegyük ezek maximumát!

```
Robot(n,x,y,L,m,u):
  darab:=(0,…,0); m:=0; u:=0
  Ciklus i=1-től n-ig
    Számol(i,m,u)
  Ciklus vége
  Ciklus i=1-től n-ig
    Ha p[i].x+p[i].y≤L+2 és tav[i]>m akkor m:=tav[i]; u:=m
  Ciklus vége
  kiir(m,u)
Eljárás vége.

Számol(i):
  Ha darab[i]=0
     akkor d:=0; honnan[i]:=0
           Ciklus j=1-től n-ig
             Ha i≠j és x[j]≤x[i] és y[j]≤y[i]
                akkor Számol(j)
                      Ha darab[j]>d
                         akkor d:=darab[j]; honnan[i]:=j
           Ciklus vége
           darab[i]:=d+1
Eljárás vége.

Kiír(m,u):
  Ha honnan[u]=0 akkor Ki: m
              különben Kiír(m,honnan[u])
  Ki: u
Eljárás vége.
```

5. feladat: Szerviz (40 pont)

Állítsunk elő egy mélységi feszítőfát, aminek majd az egyik levélleméből fogunk tovább indulni!

```
Bejár(i,levél):
  szín[i]:=szürke; db:=0
  Ciklus p∈gráf[i]
    Ha gráf[p]>0
      akkor Ha szín[p]=fehér
               akkor db:=db+1; honnan[p]:=i; Bejár(p,levél)
            különben ha honnan[i]≠gráf[p] {visszaél törlése}
                        akkor s:=gráf[p]; gráf[p]:=-s
                              Ciklus q∈gráf[s]
                                Ha q=i akkor gráf[q]:=-gráf[q]
                              Ciklus vége
  Ciklus vége
  szín[i]:=fekete
  Ha db=0 akkor levél:=i
Eljárás vége.
```

Ezután címkézzük a gráf pontjait 0,1,2 számokkal, a 0 szomszédjai 1-esek, az 1 további szomszédjai 2-esek, azoké 0-k!

```
Címkéz(i,c)
  címke[i]:=c; szín[i]:=szürke; k[c]:=k[c]+1
  Ciklus p∈gráf[i]
    Ha gráf[p]>0
       akkor Ha szín[gráf[p]]=fehér
             akkor Címkéz(gráf[p],(c+1) mod 3)
  Ciklus vége
  szín[i]:=fekete
Eljárás vége.
```

A fő eljárásban azon címkéjű pontokat kell kiírni, amelyekből `n div 3`-nál kevesebb van!

```
Szerviz(n,gráf):
  szín:=(fehér,…,fehér); honnan:=(0,…,0)
  Bejár(1,levél)
  szín:=(fehér,…,fehér); k[0]:=0; k[1]:=0; k[2]:=0
  Címkéz(levél,0)
  Ha k[0]>n div 3 akkor kiir(1,2)
  különben ha k[1]>n div 3 akkor kiir(0,2)
  különben kiir(0,1)
Eljárás vége.
```