1. feladat: Utak (50 pont)

Az alábbi ábrákon sorszámozott települések és közöttük vezető utak láthatók. Egyes településekből indulva el lehet jutni az összes településre úgy, hogy közben egy települést sem érintünk kétszer. Add meg, melyek ezek a települések!

A. B. 

C. 

2. feladat: Mit csinál? (60 pont)

Az alábbi algoritmus bemenetként kapja a K (K>1) és N (N>K) értékeket, valamint N darab egész számot az X[1]...X[N] tömbelemekben.

D:=0; C:=0; A:=1; B:=K  
Ciklus i=1-től K-ig  
 Ha X[i] páros akkor D:=D+1  
Ciklus vége  
C:=D  
Ciklus i=K+1-től N-ig  
 Ha X[i] páros akkor D:=D+1  
 Ha X[i-K] páros akkor D:=D-1  
 Ha D>C akkor C:=D; A:=i-K+1; B:=i {\*}  
Ciklus vége

A. Mi lesz C, A, B értéke, ha K=3, N=8, X=[3,2,1,2,2,1,4,4]?

B. Mi lesz C, A, B értéke, ha K=5, N=8, X=[3,2,1,2,2,1,4,4]?

C. Hogyan függ D értéke a bemenettől a \*-gal jelölt ponton?

D. Fogalmazd meg általánosan, hogyan függ C, A, B értéke a bemenettől!

E. Mi lenne az eredmény, ha megengednénk a K=1 esetet?

3. feladat: Törpék (50 pont)

A hét törpe elhatározta, hogy számítógépet fognak játszani, de nem ért minden törpe mindenhez. Morgó és Hapci fájlból tud olvasni (a fájlok neve egyik, illetve másik), Tudor hasonlítani tud, Szundi és Kuka pedig csak fájlba tud írni. Hapci, Morgó és Tudor folyamatosan figyelnek valamilyen jelzőberendezést (kezdetben mindkét jelző tilos), és ha kell, dolgoznak, ezzel szemben Kuka, Szundi, Vidor és Szende csak akkor dolgozik, ha felszólítást kap (minden felszólításra elindul a programjuk), a felszólító megvárja, hogy végezzenek. Használnak a jelzőkön kívül két közös változót (A és B). A két fájlban az adatok növekvő sorrendben vannak.

A programjuk így néz ki:

Hapci:  
 Ciklus amíg nem kész1  
 Várj amíg jelző1 szabad  
 Ha van adat(egyik) akkor Olvas(egyik, A)  
 jelző1 legyen szabad  
 különben kész1 legyen igaz  
 Ciklus vége  
 Eljárás vége.

Morgó:  
 Ciklus amíg nem kész2  
 Várj amíg jelző2 szabad  
 Ha van adat(másik) akkor Olvas(másik,B);  
 jelző2 legyen szabad  
 különben kész2 legyen igaz  
 Ciklus vége  
 Eljárás vége.

Tudor:  
 Várj amíg jelző1 tilos vagy jelző2 tilos  
 Ciklus nem kész1 és nem kész2  
 Ha A<B akkor jelző1 legyen tilos  
 különben ha A>B akkor jelző2 legyen tilos  
 különben Hívd Kukát dolgozni  
 Várj amíg jelző1 tilos és nem kész1 vagy  
 jelző2 tilos és nem kész2  
 Ciklus vége  
 Eljárás vége.

Kuka:  
 Ír(harmadik,A); jelző1 legyen tilos; jelző2 legyen tilos  
 Eljárás vége.

A. Mi lesz a harmadik-ban, ha egyik=(3,5,7,11,13) és másik=(1,5,8,11,13)?

B. Mi lesz a harmadik-ban, ha egyik=(2,4,6,8,10) és másik=(2,4,8,16)?

C. Fogalmazd meg általánosan, hogy a két fájl tartalmától függően mi kerül a harmadikba!

Átírjuk Tudor és Kuka programját, és kap munkát Szende, Szundi és Vidor is:

Tudor:  
 Várj amíg jelző1 tilos vagy jelző2 tilos  
 Ciklus amíg nem kész1 vagy nem kész2  
 Ha A<B akkor Hívd Kukát dolgozni   
 különben ha A>B akkor Hívd Szundit dolgozni  
 különben Hívd Kukát dolgozni; jelző2 legyen tilos   
 Várj amíg jelző1 tilos és nem kész1 vagy  
 jelző2 tilos és nem kész2  
 Ciklus vége  
 Ha kész1 akkor Hívd Szendét dolgozni  
 Ha kész2 akkor Hívd Vidort dolgozni  
 Eljárás vége.

Kuka:  
 Ír(harmadik,A); jelző1 legyen tilos  
 Eljárás vége.

Szundi:  
 Ír(harmadik,B); jelző2 legyen tilos  
 Eljárás vége.

Vidor:  
 Ciklus amíg nem kész1  
 Várj amíg jelző1 tilos  
 Hívd Kukát dolgozni  
 Ciklus vége  
 Eljárás vége

Szende:   
 Ciklus amíg nem kész2  
 Várj amíg jelző2 tilos  
 Hívd Szundit dolgozni  
 Ciklus vége  
 Eljárás vége

D. Mi lesz a harmadik-ban, ha egyik=(3,5,7,11,13) és másik=(1,5,8,11,13)?

E. Mi lesz a harmadik-ban, ha egyik=(2,4,6,8,10) és másik=(4,8,16)?

F. Fogalmazd meg általánosan, hogy a két fájl tartalmától függően mi kerül a harmadikba! Mi lenne, ha Szende vagy Vidor nem dolgozna?

4. feladat: Mérőkannák (70 pont)

Egy gazdának két kannája van, az egyik ***A*** literes, a másik pedig ***B***. Szeretne kimérni pontosan ***L*** liter vizet. Kezdetben mindkettő üres. Az alábbi műveleteket lehet végezni a kimérés során:

* Az A-literes kanna teletöltése
* A B-literes kanna teletöltése
* Az A-literes kanna kiürítése
* Az B-literes kanna kiürítése
* Áttöltés az A-literesből a B-literesbe (amíg az tele nem lesz, ill. van A-ban)
* Áttöltés a B-literesből az A-literesbe (amíg az tele nem lesz, ill. van B-ben)

A. Minimum hány öntéssel tud kimérni 6 liter vizet, ha az egyik kanna 10, a másik pedig 13 literes? Adj is meg egy lehetséges lépéssort!

B. Minimum hány öntéssel tud kimérni 9 liter vizet, ha az egyik kanna 10, a másik pedig 13 literes? Adj is meg egy lehetséges lépéssort!

C. Minimum hány öntéssel tud kimérni 2 liter vizet, ha az egyik kanna 5, a másik pedig 8 literes? Adj is meg egy lehetséges lépéssort!

D. Minimum hány öntéssel tud kimérni 6 liter vizet, ha az egyik kanna 5, a másik pedig 8 literes? Adj is meg egy lehetséges lépéssort!

E. Minimum hány öntéssel tud kimérni 5 liter vizet, ha az egyik kanna 6, a másik pedig 7 literes? Adj is meg egy lehetséges lépéssort!

F. Minimum hány öntéssel tud kimérni 3 liter vizet, ha az egyik kanna 6, a másik pedig 7 literes? Adj is meg egy lehetséges lépéssort!

G. Minimum hány öntéssel tud kimérni 2 liter vizet, ha az egyik kanna 6, a másik pedig 7 literes? Adj is meg egy lehetséges lépéssort!

5. feladat: Keresőfa (62 pont)

Egy keresőfa minden csomópontjára igaz, hogy tőle balra csak nála kisebb értékű elemek vannak, jobbra pedig nagyobbak, ahogy az ábrán látható.

Keresőfából elem törlését úgy kell elvégezni, hogy ez a tulajdonság megmaradjon. Három elvileg különböző esetet különböztetünk meg:

|  |  |  |
| --- | --- | --- |
| 1. Levélelemet kell törölni – törölhető, a szülőjének eggyel kevesebb gyereke lesz – a 20-as törlése: |  | |
| 2. Olyan elemet kell törölni, amelyiknek csak egy gyereke van – törölhető, a szülőjéhez az ő eddigi egyetlen gyereke kerül a helyére – a 20-as törlése: |  | |
| 3. Olyan elemet kell törölni, amelynek két gyereke van – a jobboldali ág legbaloldalibb leszármazottja kerül a helyére, de ha annak volt jobb-gyereke, az kerül az ő helyére – a 40-es törlése: |  | |
| A. Rajzold le, hogy hogyan néz ki a törlés után ez a keresőfa, ha a törlendő elem:  A1. 28  A2. 25  A3. 45  A4. 30  B. Rajzold le lépésenként, hogy hogyan néz ki a törlés után ez a keresőfa, ha a törlendő elemeket ilyen sorrendben adjuk meg:  B1. 22, 20, 40  B2. 30, 40, 35 | |  |

6. feladat: Pékség (48 pont)

Egy pékségben N polcra tesznek ki egy-egy kenyeret. Az eladó valamilyen ütemben dolgozik, azaz üres polcra tesz kenyeret, ha van. Jönnek vásárlók, akik vagy le tudnak venni egy polcról kenyeret, vagy várakozniuk kell, mert minden polc üres. A még nem olvasott üzenetek egy halmazban várakoznak, azaz közülük tetszőleges „vehető”.

A kétféle szereplő kezeléséhez kétféle eljárást használunk:

küld(vásárló,polc) – jelzi a vásárlóknak, hogy a polc sorszámú polcon van kenyér

küld(eladó,polc) – jelzi az eladónak, hogy a polc sorszámú polcon nincs kenyér

vesz(vásárló,polc) – vár egy jelzésre, hogy egy polc kiürült, a polc paraméterben kapja ennek sorszámát

vesz(eladó,polc) – vár egy jelzésre, hogy van kenyér, a polc paraméterben kapja ennek sorszámát

Kezdetben az üzletvezető, mint vásárló ezt a ciklust hajtja végre:

Ciklus i=1-től N-ig  
 küld(eladó,i)  
Ciklus vége

Az eladó programja:

Ciklus  
 kihoz egy kenyeret a raktárból {E1}  
 vesz(vásárló,m) {E2}  
 a kenyeret az m. polcra teszi {E3}  
 küld(vásárló,m) {E4}  
Ciklus vége

A vásárlók programja (többen is lehetnek, mind ezt hajtják végre):

Ciklus  
 vesz(eladó,m) {V1}  
 elveszi a kenyeret az m. polcról {V2}  
 küld(eladó,m) {V3}  
 elteszi a kenyeret {V4}  
Ciklus vége

A fenti két eljárás helyes és optimális működést eredményez. Milyen hibát vagy nem optimális működést okozna, ha az eladó tevékenység sorrendje az alábbi lenne?

A. E1, E3, E2, E4

B. E1, E2, E4, E3

C. E2, E1, E3, E4

Milyen hibát vagy nem optimális működést okozna, ha a vásárló tevékenység sorrendje az alábbi lenne?

D. V1, V3, V2, V4

E. V1, V2, V4, V3

F. V2, V1, V3, V4

7. feladat: Sorozat (60 pont)

Az alábbi algoritmus egy sorozat értékeit számolja ki:

a[1]:=1  
Ciklus i=1-től n-1-ig  
 Ha valami(i,i) akkor a[i+1]:=a[i]+2 különben a[i+1]:=a[i]+1  
Ciklus vége

A számítás használja a valami, logikai értékű függvényt:

valami(m,n):  
 Ha m<1 akkor valami:=hamis  
 különben ha a[m]=n akkor valami:=igaz  
 különben valami:=valami(m-1,n)  
 Függvény vége.

A. Mi kerül az a vektorba, ha n=10?

B. Mi a feladata a valami(m,n) függvénynek?

C. A valami függvény kiszámítása hatékonyabbá tehető egyetlen elágazás feltétel megváltoztatásával. Melyik feltétel és mire cserélendő? Miért?

D. A valami függvény rekurzió helyett ciklussal is megoldható az alábbi struktúrában. Egészítsd ki, hogy hatása azonos legyen a rekurzív megvalósítással!

valami(m,n):  
 Ciklus amíg \* {\*}  
 \*{\*\*}  
 Ciklus vége  
 Ha \* akkor valami:=hamis {\*\*\*}  
 különben valami:=igaz  
Függvény vége.