CEOI 2022

CENTRAL EUROPEAN OLYMPIAD IN INFORMATICS

VARAZDIN, CROATIA, JULY 24 - 30

2. nap

2022. július 28.

Feladatok [HUN]

Feladat	Időkorlát	Memóriakorlát	Pontszám
Drawing	1.5 mp	512 MiB	100
Measures	1.5 mp	512 MiB	100
Parking	2 mp	512 MiB	100
Összesen			300

REPUBLIC OF CROATIA
Ministry of Science and Education

CROATIAN ASSOCIATION OF TECHNICAL CULTURE

CROATIAN COMPUTER SCIENCE ASSOCIATION

Drawing

A *Festés & Bor* az első zágrábi festőstúdió, amely pihentető és relaxáló festőórákat kínál, egy-egy pohár borral segítve a mű megalkotását. Az óra alatt a diákok egy témát kapnak és a festőmesterek segítségével általában sikerül is lenyűgöző alkotást festeniük.

Ante egy festőmester, Luka a tanítványa. Ez a feladat egy történetet mesél el egy olyan óráról, amelyen a szokásosnál egy kicsit több bor fogyott el.

Ante: "Fess nekem egy fát!"

Luka: "Rendben. Milyen fát szeretnél? Pálmát, tölgyet, fenyőfát...?

Ante: "Egy irányítatlan, összefüggő, körmentes gráfot akarok!"

Luka: "Azt meg tudom csinálni... Valami más kívánság?"

Ante: "Azt szeretném, ha egyetlen csúcs sem lenne összekötve háromnál több másik csúccsal!"

Luka: "Uhm, Rendben... Nos, sok ilyen fa létezik."

Ante: "Itt egy lista az élekről, nekem ilyen kellene!"

Luka: "Oké, hűha. Úgy látom, hogy többféleképpen is meg lehet rajzolni."

Ante: "Itt van a sík azon pontjainak listája, ahová a csúcsokat szeretném, ha tennéd. Nem szeretném, hogy legyen a rajzon két egymást metsző él."

Luka: "Megoldom!"

A feladatod az, hogy segíts Lukának lerajzolni a fát, Ante kívánságai szerint. Pontosabban, adott egy fa leírása, úgy, hogy egyetlen csúcs sem szomszédos több, mint három másik csúccsal, és adott a síkon lévő pontok listája. Keress egy-az-egyhez kapcsolatot a csúcsok és a pontok között úgy, hogy amikor a fa éleit a megadott pontokat összekötő egyenesekként rajzoljuk meg, akkor azok nem metszik egymást (kivéve a csúcsokban).

Bemenet

A bemenet első sora az N egész számot tartalmazza, a fa csúcsainak a számát, illetve a síkban megadott pontok számát.

Az ezt követő $N-1$ sor a fa éleit írja le, soronként egyet. Minden él két egész számmal van leírva: a és b, a csúcsok sorszáma, amik közt élet kell húzni. A csúcsok 1-től N-ig vannak sorszámozva.

Azt biztosan tudjuk, hogy minden csúcs legfeljebb három másikkal van összekötve.

A következő N sorban a fa rajzolásához használandó pontok vannak leírva, minden sorban egy. Minden pont két koordinátával van leírva, melyek egész számok. Nincs két pont, amiknek ugyanazok a koordinátái és **semmelyik három pont nem esik egy egyenesre**.

Kimenet

A kimenet az $1, 2, \ldots, N$ egész számok egy permutációja, egy sorba írva, szóközzel elválasztva. Az i-edik szám annak a csúcsnak a sorszáma, ami a bemenetben az i-edik pont a síkon.

Ha több jó megoldás létezik, akkor bármelyik kiírható.

Garantált, hogy mindig létezik jó megoldás.

Pontozás

Minden részfeladatban a pontok koordinátái egész számok 0 és 10^9 között.

Részfeladat	Pontszám	Korlátok
1	10	$3 \leq N \leq 200\,000$, a megadott pontok megfelelő sorrendben konvex sokszöget alkotnak.
2	15	$1 \leq N \leq 4\,000$
3	15	$1 \leq N \leq 10\,000$
4	35	$1 \leq N \leq 80\,000$
5	25	$1 \leq N \leq 200\,000$

Példák

input

```
3
1 2
2 3
10 10
10 20
20 10
```

output

```
1 2 3
```

input

```
5
1 2
1 3
1 4
4 5
10 10
10 30
30 10
30 30
20 25
```

output

```
5 4 2 3 1
```

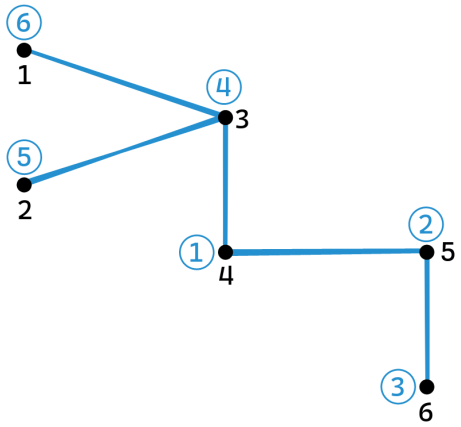
input

```
6
1 2
2 3
1 4
4 5
4 6
10 60
10 40
40 50
40 30
70 30
70 10
```

output

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

A harmadik példa magyarázata:



A kék számok jelentik a csúcssorszámokat, míg a fekete számok a pontok sorszámait.

Measures

A COVID-19 világjárvány több szempontból is meglepte az egész világot. Egyik napról a másikra az embereknek világszerte új életmódhoz kellett alkalmazkodniuk, a helyi hatóságok megelőző intézkedéseit kellett betartani, melyeknek a betegség terjedésének visszaszorítása és szabályozása volt a céljuk.

Egy távoli jövőbeli, valószínűtlen, ám pusztító járványra készülve, a Horvát Nemzeti Közegészségügyi Intézet úgy döntött, hogy különböző kutatási részlegeket hoz létre. Ezeknek a fő célja olyan rendkívül hatékony protokollok kidolgozása, amelyek segítik a lakosságot egy új, megelőző intézkedés gyors betartásában.

Alenka az egyik ilyen részlegen dolgozik, és egy olyan forgatókönyvet vizsgál, amelyben emberek egy csoportja áll egy sorban, pl. a postahivatal előtt, és hirtelen új biztonsági intézkedés lép életbe, amely előírja, hogy bármely két ember közötti távolságnak legalább D-nek kell lennie.

Egy olyan alkalmazást már megvalósított, amely lehetővé teszi a felhasználó számára, hogy megadjon egy D távolságot és N ember helyét koordináták formájában egy egyenesen. Az alkalmazás ezután megrajzolja a helyzetet reprezentáló vonal képét, és kiszámítja - másodpercben kifejezve - a legkisebb t_{opt} időt, ami szükséges ahhoz, hogy az emberek elérjenek egy olyan új elrendezést, amely kielégíti a megelőző intézkedést. Az alkalmazás feltételezi, hogy az emberek azonnal elkezdenek optimálisan átrendeződni és minden ember ugyanolyan állandó, egységnyi sebességgel mozog másodpercenként.

Most egy új funkciót szeretne hozzáadni, amely lehetővé teszi a felhasználó számára, hogy további M embert hozzáadjon a csoporthoz a vonalra koppintva, megadva a helyüket. Az alkalmazásnak újra kell számolnia a t_{opt} értéket minden egyes koppintás, azaz minden egyes személy csoporthoz adása után!

A feladatod segíteni Alenkának az új funkció megvalósításában.

Bemenet

A bemenet első sora három pozitív egész számot tartalmaz: N-et, M-et és D-t a feladat leírásából.

A második sorban N darab egész szám szerepel: $a_1, \ldots, a_N$, az eredetileg sorban álló N ember helyzete.

A harmadik sorban M darab egész szám szerepel: $b_1, \ldots, b_M$, az M hozzáadott ember helyzete.

Kimenet

A kimenet M számot tartalmazzon egy sorban. Ezekből az i-edik a t_{opt} azon értéke, ami az $(N+i)$ darab, kezdetben a_1, a_2, …, a_N, b_1, …, b_i helyzetű ember optimális átrendeződéséhez szükséges.

A kiírásban minden szám tizedes alakban, tizedesvessző helyett pontot használva, a tizedesben a végén nullák nélküli, legrövidebb alakban legyen megadva. Például a kimenetben `1.23` a `1.2300` helyett, és `123` in `123.` vagy a `123.0` helyett is. Bizonyított, hogy a válasz mindig kiírható véges decimális számként.

Pontozás

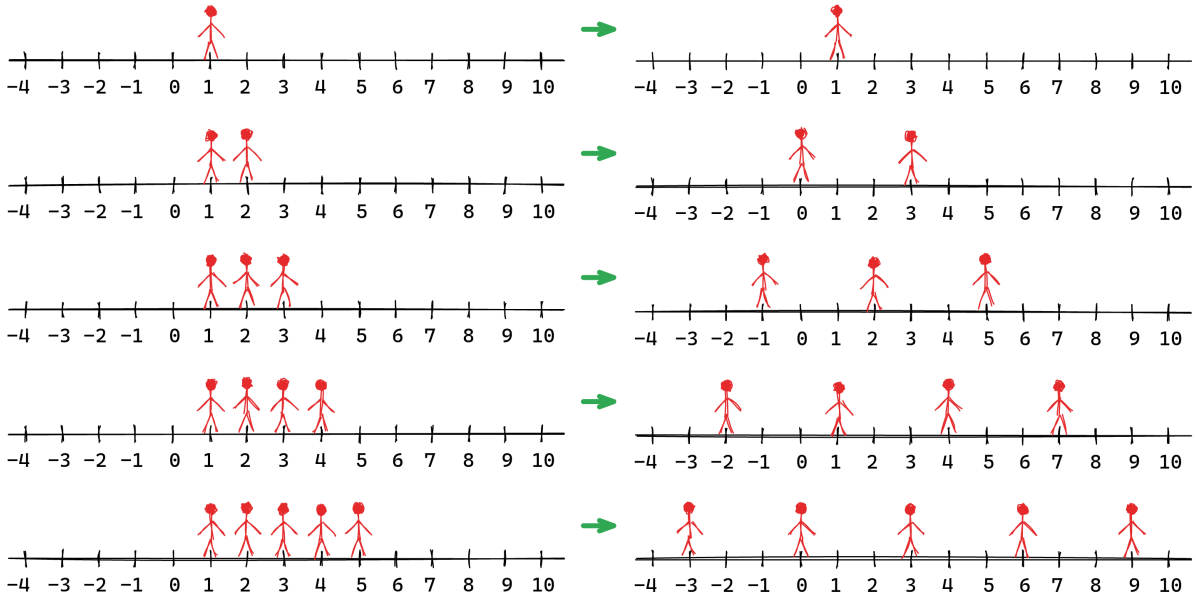
Minden részfeladatban $1 \leq D, a_1, \ldots, a_N, b_1, \ldots, b_M \leq 10^9$.

Részfeladat	Pontszám	Korlátok
1	10	$0 \leq N \leq 2\,000$, $1 \leq M \leq 10$
2	14	$0 \leq N \leq 200\,000$, $1 \leq M \leq 10$
3	35	$N = 0$, $1 \leq M \leq 200\,000$, $b_1 \leq \cdots \leq b_M$
4	41	$N = 0$, $1 \leq M \leq 200\,000$

Példák

input	input	input
2 1 2 1 3 2	0 5 3 1 2 3 4 5	3 3 3 3 3 3 3 3 3
output	**output**	**output**
1	0 1 2 3 4	4.5 6 7.5

A második példa magyarázata:



Parking

Valerija parkolósegédként dolgozik egy előkelő étteremben. A munkája abból áll, hogy az érkező vendégeket illendően köszönti, majd elkéri a kocsikulcsukat és leparkolja az autójukat a közeli parkolóban. Miután végeztek a vendégek, gondoskodik róla, hogy mindenki épségben visszakapja a saját járművét és elégedetten távozzon a helyszínről.

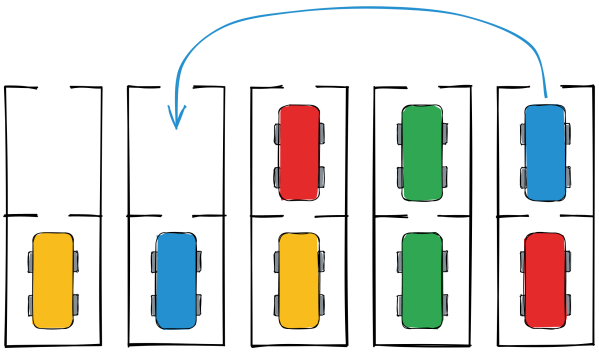
Egy este, miután az összes autót leparkolta, egy rendkívül érdekes dolgot vett észre: a parkolóban lévő, összesen $2N$ darab gépkocsi mindegyike N darab szín valamelyikével van befestve, ráadásul mindegyik színhez pontosan két autó tartozik. Az N-féle színt a továbbiakban az 1 és N közti egészekkel jelöljük.

A parkoló M darab szomszédos parkolóhelyet tartalmaz, melyeket 1-től M-ig sorszámozunk. Minden parkolóhelyen legfeljebb két autó parkolhat, de a parkolóhelynek csak egyetlen bejárata van. A bejárathoz közelebb álló járművet *felső autónak*, a távolabb állót *alsó autónak* nevezzük. Az alsó autó nem tudja elhagyni a parkolóhelyet, ha ott felső autó is tartózkodik. Valerija úgy parkolja le az autókat, hogy minden parkolóhely

- vagy üres,
- vagy pontosan egy autót tartalmaz, ami alsó autó,
- vagy tele van, azaz alsó és felső autót is tartalmaz.

Valerija szeretné átrendezni az autókat úgy, hogy minden azonos színű gépkocsipár ugyanazon a parkolóhelyen álljon. Nem számít, hogy melyik parkolóhelyen melyik színű pár fog állni, vagy hogy a két azonos színű autó közül melyik lesz a felső és melyik az alsó autó. Az átrendezést *átparkolások* sorozatával szeretné megvalósítani. Minden átparkolás során kiválaszt egy autót, ami képes elhagyni a jelenlegi parkolóhelyét, és átáll vele egy olyan parkolóhelyre, ami

- vagy üres: ekkor a kiválasztott autót alsó autóként parkolja le;
- vagy egyetlen, a **kiválasztott autóval azonos színű autót** tartalmaz: ekkor a kiválasztott autót felső autóként parkolja le.



Az első példában szereplő elrendezés, az egyetlen szabályos átparkolással.

Valerija szeretné minimalizálni az átparkolások számát. Feladatod, hogy segíts neki megtalálni a legrövidebb átparkolássorozatot, mely eredményeként minden azonos színű gépkocsipár azonos parkolóhelyre kerül, vagy jelezd, hogy ilyen sorozat nem létezik.

Bemenet

Az első sor két, szóközzel elválasztott egész értéket tartalmaz, N és M értékét.

A következő M sor közül az i-edik két, szóközzel elválasztott egész értéket tartalmaz, az i-edik parkolóhely leírását. Az első szám, b_i ($0 \leq b_i \leq N$) a parkolóhelyen álló alsó autó színét adja meg, ha a parkolóhely tartalmaz alsó autót; egyébként b_i értéke 0. A második szám, t_i ($0 \leq t_i \leq N$) a parkolóhelyen álló felső autó színét adja meg, ha a parkolóhely tatalmaz felső autót; egyébként t_i értéke 0. Garantált, hogy ha a parkolóhely nem tartalmaz alsó autót, akkor nem tartalmaz felső autót sem, azaz ha $b_i = 0$, akkor $t_i = 0$ is teljesül.

Kimenet

Ha nem létezik átparkolások olyan sorozata, mely teljesíti Valerija célkitűzését, akkor a programodnak egyetlen sorba -1-et kell kiírnia.

Egyébként a kimenet első sorába K értéke kerüljön, ahol K a legrövidebb átparkolássorozat hossza, amivel teljesíthető a célkitűzés.

A következő K sor közül az i-edik sorba az i-edik átparkolás leírását kell kiírni. Pontosabban, az i-edik sorba két egész szám, x_i és y_i kerüljön ($1 \leq x_i, y_i \leq M, x_i \neq y_i$), ahol Valerijának az x_i parkolóhelyen álló, azt elhagyni képes autóval kell átparkolnia az y_i parkolóhelyre. Értelemszerűen, az átparkolás végrehajtása előtt az x_i parkolóhelyen legalább egy autónak kell állnia, és az x_i parkolóhelyet elhagyó autónak képesnek kell lennie átparkolni az y_i parkolóhelyre (vagyis y_i vagy üres, vagy egyetlen, az átparkoló autóval azonos színű autót tartalmaz!).

Pontozás

Minden részfeladatban $1 \leq N \leq M \leq 200\,000$ teljesül.

Ha a programod helyesen meghatározza és kiírja K értékét egy részfeladat összes tesztesetére, de az átparkolások leírását hibásan adja meg (vagy egyáltalán nem írja ki), akkor a részfeladatra szerezhető pontok 20%-át kapja meg.

Részfeladat	Pontszám	Korlátok
1	10	$M \leq 4$
2	10	$2N \leq M$
3	25	$N \leq 1\,000$ és kezdetben minden parkolóhely vagy üres, vagy tele van.
4	15	Kezdetben minden parkolóhely vagy üres, vagy tele van.
5	25	$N \leq 1\,000$
6	15	Nincsenek további korlátok.

Példák

input

```
4 5
1 0
2 0
1 3
4 4
3 2
```

output

```
3
5 2
3 5
3 1
```

input

```
4 5
0 0
2 1
3 1
3 4
2 4
```

output

```
-1
```

input

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

output

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

Az első példa magyarázata: A feladatleírásban szereplő ábra a parkolóhelyek kezdeti állapotát mutatja. Látható, hogy ebben a példában az átparkolások egyértelműek, vagyis az első és a második átparkolás egyetlen megengedett módon történhet csak, a harmadik átparkolásra pedig két ekvivalens lehetőség kínálkozik, és mindkettő teljesíti a célkitűzést.