

Sakktábla bábukkal

Péter öccse talált a padláson egy csomó játékgurát. Elhatározta, hogy sakkozni fog a figurákkal, azonban van egy kis bökkenő: nem tud sakkozni, és még sakktáblája sincs. Ezért felrajzolt egy papírlapra egy 8×8 -as négyzetrácsot, melynek minden mezőjét világosra, vagy sötétre színezte (nagyjából ennyiben merültek ki a sakkkal kapcsolatos ismeretei). Ezután figurákat helyezett el a mezők egy részére.

Péter a szobába belépve ránézett a kisöccse által előkészített játéktáblára. A 8×8 mező mindegyikére

- ha a mezőn nincsen figura, akkor Péter látja a mező színét (világos vagy sötét);
- ha a mezőn figura áll, akkor Péter nem látja a mező színét.

Péter ekkor elgondolkozott, hogy az öccse egyáltalán szabályosan színezte-e ki a tábla mezőit, azaz minden sorban és oszlopban váltakozik-e a mezők színe.¹ Írj programot, ami eldönti, hogy a látható színű mezők alapján lehetséges-e, hogy a mezők a sakktáblának megfelelően vannak kiszínezve!

Bemenet

A standard bemenet összesen 8 sorból áll, és soronként egy 8 hosszú karakterláncot tartalmaz. Az i -edik sor j -edik karaktere ($T_{i,j}$) megadja, hogy az i -edik sor j -edik mezőjén mit lát Péter:

- F: a mezőn figura áll;
- V: a mező világos színű;
- S: a mező sötét színű.

Kimenet

A standard kimenetre egy sort kell írni, mely a LEHETETLEN szöveget tartalmazza, ha a mezők színezése a látottak alapján biztosan nem lehet szabályos sakktábla színezés, vagy a LEHETSEGES szöveget, ha lehet szabályos a színezés.

Példa

Bemenet

```
SVSVSVSV
FFFFFFFF
FFFFFFFF
FFFFFFFF
FFFFFFFF
FFFFFFFF
FFFFFFFF
FFFFFFFF
```

Kimenet

```
LEHETSEGES
```

¹A bal alsó mezőnek itt nem kell sötétnek lennie, szemben a sakk szabályai szerint elhelyezett sakktáblával.

Bemenet

```
VFFFFFFF
FFFFFFF
FFFFFFF
FFFFFFF
FFFFFFSF
FFFFFFF
FFFFFFF
FFFFFFF
FFFFFFF
```

Kimenet

LEHETETLEN

Korlátok

$T_{i,j} \in \{F, V, S\}$ minden $i, j = 1 \dots 8$ -ra

Időlimit: 0.5 mp.

Memórialimit: 256 MB

Pontozás

A megoldásodat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	az első sor egyik mezőjén sincs figura, a többi sor minden mezőjén figura áll	16
2	egyik mezőn sincs figura	14
3	pontosan két mező színe látszódik	22
4	nincsenek további megkötések	28

Munkák

Egy vállalkozó a következő N munkanapra összesen M megbízást kapott. Minden megbízás egy konkrét A_i napra szól és a teljesítése H_i bevételt eredményez. Ugyanarra a napra több megbízása is lehet a vállalkozónak, de egy nap csak egy megbízás teljesítésére van elég ideje, sőt, a megbízások teljesítése között legalább egy nap szünetet kell tartania. Az N -edik napon végzett munka után nem kell szünetet tartani.

Írj programot, ami meghatározza, hogy melyik megbízásokat kell a vállalkozónak teljesítenie ahhoz, hogy a bevétele a lehető legnagyobb legyen!

Bemenet

A standard bemenet első sora két egész számot tartalmaz, a napok N számát és a megbízások M számát.

A következő M sor mindegyike két egész számot tartalmaz, az első szám a megbízás napjának az A_i sorszáma, a második pedig a teljesítésével szerzhető H_i bevétel értéke. A megbízásokat a bemenetbeli sorrendben azonosítjuk az $1, \dots, M$ számokkal (az $i + 1$ -edik sorban lévő megbízás azonosítója i).

Kimenet

A standard kimenet első sorába két egész számot kell írni, az első szám az elérhető legnagyobb bevétel értéke, a második pedig az elvégzésre kiválasztott munkák K darabszáma legyen! A második sorba a kiválasztott K munka sorszámát kell írni, tetszőleges sorrendben.

Több megoldás esetén bármelyik megadható.

Példa

Bemenet

10 7

3 1

1 8

3 2

2 1

4 10

6 6

5 8

Kimenet

24 3

2 5 6

Korlátok

$$2 \leq N \leq 100\,000$$

$$1 \leq M \leq 100\,000$$

$$1 \leq A_i \leq N \text{ minden } i = 1 \dots M\text{-re}$$

$$1 \leq H_i \leq 10\,000 \text{ minden } i = 1 \dots M\text{-re}$$

Időlimit: 0.6 mp.

Memórialimit: 64 MB

Pontozás

A megoldásodat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	$N \leq 10, M \leq 100$, minden munka haszna ugyanaz	10
2	$N \leq 100, M \leq 10$, a munkák haszna két különböző érték valamelyike	10
3	$N \leq 10, M \leq 100$	10
4	$N \leq 10\,000, M \leq 10$	10
5	$N \leq 10\,000, M \leq 100$	10
6	$N, M \leq 50\,000$	10
7	nincsenek további megkötések	20

Minden részfeladatban **részpontokat** lehet szerezni: a részfeladat pontszámának 60%-a jár, ha az első kiírt szám (a maximális bevétel értéke) helyes a részfeladathoz tartozó összes tesztesetben. A részpontok megszerzéséhez ügyelj arra, hogy a megoldásod

- vagy egyáltalán ne írjon ki K értéket, hanem egyetlen sorba a bevétel legnagyobb lehetséges értékét írja csak ki;
- vagy ha a kimenetének első sorában szerepel a K érték, akkor a második sorba pontosan K darab egész értéket írjon ki (K lehet 0 is).

Ha a fentiek egyike sem teljesül, akkor előfordulhat, hogy az értékelő formai hiba miatt nem adja meg a részpontokat.

Hírlánc

Anna osztálya a téli szünetre hírláncot képzett, hogy szórakoztassák magukat: minden tanuló kiválasztott pontosan egy másik tanulót, akinek továbbít minden üzenetet, amelyet vagy ő indított, vagy másoktól kapott. Ha egy üzenet visszaér valakihez, akkor ő már nem továbbítja azt ismét.

Írj programot, ami kiszámítja, hogy ki az a tanuló, aki által elindított hír a legtöbb tanulóhoz eljut!

Bemenet

A standard bemenet első sora a tanulók N számát tartalmazza. A második sor pontosan N egész számot tartalmaz, az i -edik szám annak a tanulónak a C_i sorszáma, akinek az i sorszámu tanuló továbbítja az üzeneteket. A tanulókat az $1, \dots, N$ számokkal azonosítjuk.

Kimenet

A standard kimenet első sorába két egész számot kell írni: az első szám annak a tanulónak a K sorszáma legyen, aki által indított hír a legtöbb tanulóhoz eljut. A második szám azon tanulók száma legyen (a hírt elindító tanulót is beleértve), akik megkapják a K tanuló által indított hírt.

Több megoldás esetén bármelyik megadható.

Példa

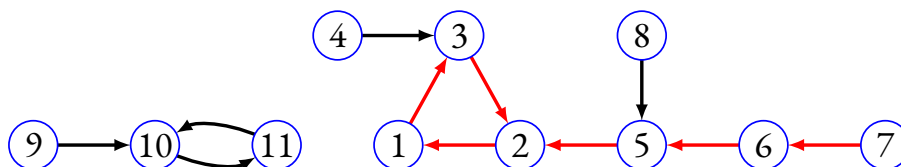
Bemenet

```
11
3 1 2 3 2 5 6 5 10 11 10
```

Kimenet

```
7 6
```

A példában szereplő hírláncot az alábbi ábra szemlélteti.



A 7-es sorszámu tanulótol jut el a hír a legtöbb tanulóhoz, összesen 6 emberhez. A hír útját piros szín jelöli ebben az esetben: amikor a 2-es sorszámu tanuló másodszer is megkapja a hírt (a 3-as tanulótol), már nem küldi tovább ismét.

Korlátok

$$3 \leq N \leq 200\,000$$

$$1 \leq C_i \leq N \text{ és } C_i \neq i \text{ minden } i = 1 \dots N\text{-re}$$

Időlimit: 0.5 mp.

Memórialimit: 256 MB

Pontozás

A megoldásodat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	$N \leq 2000$	20
2	minden tanuló pontosan egy másik tanulótól kap hírt	18
3	nincsenek további megkötések	42

Díjazás több helyezettnek

Egy programozási versenyen az első H helyezett összesen E euró jutalmat kap. A szervezők még nem döntöttek el, hogyan osszák el ezt az összeget. Úgy szeretnék szétosztani a nyereményt, hogy mindenki kapjon legalább 1 eurót és mindenki többet kapjon azoknál a versenyzőknél, akiket megelőzött.

Írj programot, ami meghatározza az összes lehetséges díjazást! A díjazásokat fordított ábécé szerinti sorrendben kell felsorolni. Ez azt jelenti, hogy az $A = (a_1, \dots, a_H)$ (ahol a_i az i -edik helyezett díja) akkor és csak akkor van előbb, mint a $B = (b_1, \dots, b_H)$ díjazás, ha $a_i > b_i$ teljesül a legkisebb olyan i indexre, amire $a_i \neq b_i$. Például $E = 14$ esetén a $(6, 5, 2, 1)$ díjazás megelőzi a $(6, 4, 3, 1)$ díjazást a felsorolásban.

Bemenet

A standard bemenet első sora két egész számot tartalmaz, a díjazottak H számát és a jutalomként kiosztandó E összeget.

Kimenet

A standard kimenetre soronként egy-egy lehetséges díjazást kell kiírni fordított ábécé szerinti sorrendben! Minden sorban H darab pozitív egész érték álljon, ahol az i -edik szám az i -edik helyezett versenyző díja.

Példa

Bemenet

4 14

Kimenet

8 3 2 1

7 4 2 1

6 5 2 1

6 4 3 1

5 4 3 2

Korlátok

$$2 \leq H \leq 19$$

$$\frac{H \cdot (H+1)}{2} \leq E \leq 200$$

A lehetséges díjazások száma legfeljebb 1000

Időlimit: 0.5 mp.

Memórialimit: 256 MB

Pontozás

A megoldásokat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	$H = 2$	5
2	$H = 3$	5
3	$H = 4$	15
4	$E \leq 50$	15
5	$E \leq 100$	15
6	nincsenek további megkötések	25

Városnézés

Bittonia történelmi városában N tér található, melyeket 1-től N -ig számoznak. Mivel sok a látogató, így a tereket egyirányú utakkal kötötték össze, oly módon, hogy ha egy teret egyszer elhagy egy látogató, akkor oda később már ne tudjon visszatérni. Minden téren van bizonyos számú látványosság (lehet 0 is). Ha egy látogató a téren van, akkor az összes ott található látványosságot meg tudja tekinteni.

Városnézést tervezel, ami az 1. térről indul és az N . téren ér véget. Írj programot, mely kiszámolja a maximális mennyiségű látványosságot amit láthatsz a városnézésed alatt, és megad egy útvonalat ehhez!

Bemenet

A standard bemenet első sorában a terek N száma és az ezeket összekötő egyirányú utak M száma található. A második sorban N darab nemnegatív szám található, az i . szám az i . téren található látványosságok P_i száma.

A következő M sor mindegyike egy utat leíró u_j, v_j számpárt tartalmaz, ami azt jelenti, hogy az u_j térről a v_j térre egyirányú út vezet.

Kimenet

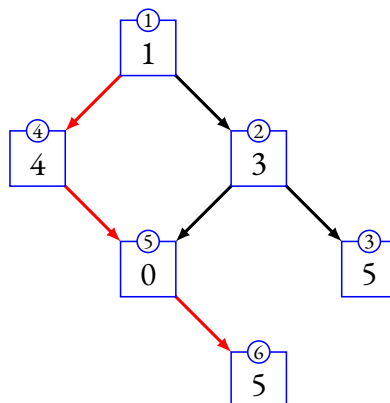
A standard kimenet első sorába egy egész számot kell kiírni, a megnézhető látványosságok számának legnagyobb lehetséges értékét. A következő sorba egy útvonalat kell kiírni, amellyel ezt el lehet érni. Több lehetséges útvonal esetén bármelyik megadható.

Ha nem lehet eljutni az 1. térről a N . térre, akkor az első és egyetlen sorba -1 -et kell kiírni.

Példa

Bemenet	Kimenet
6 6	10
1 3 5 4 0 5	1 4 5 6
1 2	
2 3	
1 4	
2 5	
4 5	
5 6	

Az alábbi ábra a példa bemenetet szemlélteti. A körökbe írt szám a tér sorszáma, a négyzetbe írt szám a látványosságok száma. Az optimális útvonalat piros szín jelöli.



Korlátok

$$2 \leq N \leq 20\,000$$

$$1 \leq M \leq 50\,000$$

$$0 \leq P_i \leq 10\,000 \text{ minden } i = 1 \dots N\text{-re}$$

$$1 \leq u_j \neq v_j \leq N \text{ minden } j = 1 \dots M\text{-re}$$

Az (u_j, v_j) párok mind különbözőek

Időlimit: 1.0 mp.

Memórialimit: 256 MB

Pontozás

A megoldásodat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	az 1. térről pontosan egy féle képpen lehet eljutni bármely másik térre	20
2	$u_i < v_i$ minden $i = 1 \dots M$ -re	25
3	$N \leq 100$ és $M \leq 1000$	20
4	nincsenek további megkötések	15