

Járdakövezés

Csongorék teljes házfelújítást végeznek, többek közt az udvari járdát is le szeretnék kövezni. A járda H centiméter hosszú. Sajnos a közeli szaküzletben csak két méretben árulnak köveket: az egyik típus A , a másik B centiméter hosszú. A köveket nem lehet elvágni. Minden kő és az építendő járda is egyforma széles.

Írj programot, ami kiszámolja, hogy az egyes típusokból hány darabot vegyenek, hogy le tudják kövezni a járdát, vagy megadja, hogy nem lehet megoldani a kövezést.

Bemenet

A standard bemenet első sora három egész számot tartalmaz: a járda H hosszát és a két típusú kő A és B hosszát.

Kimenet

A standard kimenetre két egész számot kell kiírni: hány darabot kell venni az egyik, illetve másik típusú kőből. Ha a megadott méretekkel nem lehetséges kikövezni a járdát, akkor egyetlen sorba a $0\ 0$ számpárt írd ki.

Több megoldás esetén bármelyik megadható.

Példa

Bemenet

21 6 9

Kimenet

2 1

Korlátok

$$1 \leq H, A, B \leq 10\,000\,000$$

$$A \neq B$$

Időlimit: 1.0 mp.

Memórialimit: 256 MB

Pontozás

A megoldásodat több különböző tesztesetre lefuttatjuk. A teszteseteknek önálló pontértéke van és a megoldás pontszáma a megoldott tesztesetek pontszámainak összege. A feladat összpontszáma a legtöbb pontot elért megoldás pontszáma.

A pontszám 16%-a szerezhető olyan tesztekre, ahol $H, A, B \leq 10$.

A pontszám további 16%-a szerezhető olyan tesztekre, ahol $H = A$ vagy $H = B$.

A pontszám további 32%-a szerezhető olyan tesztekre, ahol $H, A, B \leq 1000$.

Ültetés

Egy N tagú osztály Súgópostát szeretne játszani. A játék során az osztály tagjai leülnek N darab, sorban egymás mellé elhelyezett székre. Ezután az első széken ülő tanuló súg valamit a második széken ülőnek, aki továbbítja azt a harmadik széken ülőnek, és így tovább egészen az N -edik széken ülő tanulóig.

Szeretnék, hogy az eredeti üzenet változtatás nélkül érjen végig a soron, ezért minden tanuló megadta egy osztálytársát, akinek súgni szeretne. Egy tanuló kívánsága tehát pontosan akkor teljesül, ha a tanuló által megadott osztálytárs a sorban közvetlenül utána következő székre ül le.

Az osztályfőnök szeretné úgy leültetni a gyerekeket a székekre, hogy a lehető legtöbb kívánság teljesüljön. Írj programot, amely megad egy ilyen ültetést.

Bemenet

A standard bemenet első sora a tanulók N számát tartalmazza. A tanulókat az $1, 2, \dots, N$ számokkal azonosítjuk.

A második sor pontosan N darab egész számot tartalmaz, az i -edik szám annak a tanulóknak az S_i azonosítója, akinek az i azonosítójú tanuló súgni szeretne.

Kimenet

A standard kimenet első sorába azt a legnagyobb M egész számot kell írni, ahány tanulóknak a kívánsága teljesíthető megfelelő ültetéssel. A következő sorba N darab egész értéket kell írni, az i -edik szám annak a székeknek a sorszámát, amelyikre az i azonosítójú tanuló ül.

Több megoldás esetén bármelyik megadható.

Példa

Bemenet

8
3 1 4 1 1 7 8 6

Kimenet

5
2 1 3 4 8 5 6 7

Az egyes székeken ülő tanulók azonosítói sorrendben: $2, 1, 3, 4, 6, 7, 8, 5$. Tehát 5 kívánság teljesül (a 4, 8 és 5 azonosítójú tanulók kívánsága nem teljesül). Belátható, hogy ennél több kívánság egyszerre nem teljesíthető.

Korlátok

$$5 \leq N \leq 10\,000$$

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

$$S_i \neq i \text{ minden } i = 1 \dots N\text{-re}$$

Időlimit: 0.6 mp.

Memórialimit: 64 MB

Pontozás

A megoldásodat több különböző tesztesetre lefuttatjuk. A teszteseteknek önálló pontértéke van és a megoldás pontszáma a megoldott tesztesetek pontszámainak összege. A feladat összpontszáma a legtöbb pontot elért megoldás pontszáma.

A pontszám 7%-a szerezhető olyan tesztekre, ahol $N = 5$.

A pontszám további 7%-a szerezhető olyan tesztekre, ahol $N \leq 10$.

A pontszám további 7%-a szerezhető olyan tesztekre, ahol $N \leq 100$.

A pontszám további 13%-a szerezhető olyan tesztekre, ahol pontosan egy olyan tanuló van, akit senki sem választott.

A pontszám további 13%-a szerezhető olyan tesztekre, ahol nincs két olyan tanuló, akik ugyanazt a tanulót választották.

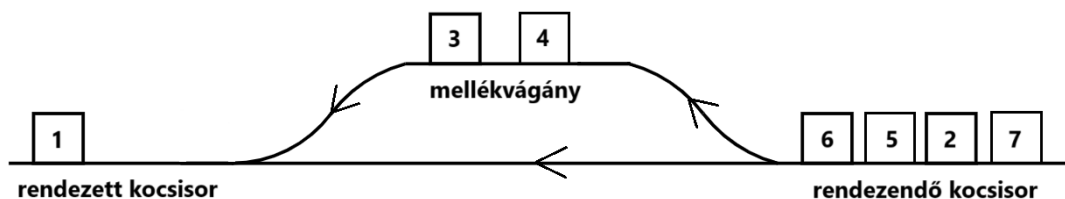
Minden **tesztesetben** részpontokat lehet szerezni. A teszteset pontszámának 50%-a jár abban az esetben, ha M értéke helyes, de a kimenet hibás ültetést tartalmaz, vagy csak az M értékből áll. Ha a program kimenete tartalmaz ültetést a második sorban, akkor ügyelj arra, hogy pontosan N darab 1 és N közti egész értéket tartalmazzon, különben az értékelő formai hiba miatt nem biztos, hogy megadja a részpontokat.

Vasúti rendezés

Egy vasúti vágányon egy sorban N darab vasúti kocsi áll. A munkások szeretnék a kocsikat egy adott sorrendbe berendezni: mindegyik kocsira ráírták, hogy a rendezett sorrendben hányadik helyen kellene lennie.

A rendezéshez egy mellékvágányt és az alábbi három művelet tudják használni:

1. Az eredeti, rendezendő sorból az **első** kocsit átviszik a rendezett sor **végére**.
2. A rendezendő sorból az **első** kocsit átviszik a mellékvágányon lévő sor **végére**.
3. A mellékvágányon lévő kocsisorból az **első** kocsit átviszik a rendezett sor **végére**.



A példában szereplő rendezés első három lépése.

Írj programot, amely kiszámítja a legnagyobb M értéket, amire teljesül, hogy az $1, \dots, M$ sorszámokat viselő kocsik ebben a sorrendben átvihetők a rendezett sorba.

Bemenet

A standard bemenet első sora a kocsik N számát tartalmazza. A második sor N darab, páronként különböző egész számot tartalmaz, ahol az i -edik szám a mostani sorban i -edik helyen álló kocsira írt S_i sorszám.

Kimenet

A standard kimenet első sorába azt a legnagyobb M értéket kell írni, amelyre teljesül, hogy az első M sorszámot viselő kocsi rendezhető a megadott módszerrel.

Példa

Bemenet

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

Kimenet

```
4
```

Az $1, 2, 3, 4$ sorszámokat viselő kocsik rendezhetők például az alábbi módon:

- A 3 majd a 4 sorszámú kocsikat a mellékvágány végére visszük.
- Az 1 sorszámú kocsit a rendezett sor végére visszük.
- A 6 majd az 5 sorszámú kocsikat a mellékvágány végére visszük.
- A 2 sorszámú kocsit a rendezett sor végére visszük.
- A 3 és 4 sorszámú kocsikat a mellékvágány elejéről a rendezett sor végére visszük.

Belátható, hogy ennél több kocsit nem lehet rendezni.

Korlátok

$$3 \leq N \leq 100\,000$$

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

Az S_i értékek páronként különbözőek

Időlimit: 0.5 mp.

Memórialimit: 256 MB

Pontozás

A megoldásodat több különböző tesztesetre lefuttatjuk. A teszteseteknek önálló pontértéke van és a megoldás pontszáma a megoldott tesztesetek pontszámainak összege. A feladat összpontszáma a legtöbb pontot elért megoldás pontszáma.

A pontszám 8%-a szereshető olyan tesztekre, ahol $N = 3$.

A pontszám további 28%-a szereshető olyan tesztekre, ahol $N \leq 10$.

A pontszám további 32%-a szereshető olyan tesztekre, ahol $N \leq 500$.

Fagyasztó

Egy sarkvidéki kutatóállomáson rekedtél, ahonnan a külvilág elérhetetlen a tomboló hóvihár miatt. Az egyetlen esélyed a túlélésre a meglévő élelmiszerkészlet beosztása.

Összesen N adag étel van a fagyasztóban. Minden ételnek ismered az S_i szavatossági idejét (az étel az S_i -edik nap végéig fogyasztható), és az M_i minőségi értékét.

Minden nap a következő szabályok szerint kell választanod:

- Két adag ételt kell elfogyasztanod.
- Mindkét ételnek még fogyaszthatónak kell lennie (az adott nap a szavatossági időn belül van).
- Az ételek minősége egyenként legalább 4.
- A két étel együttes minősége (a minőségeik összege) legalább 9.

Írj programot, ami meghatározza, hogy legfeljebb hány napig tudod biztosítani a napi étkezésedet a fagyasztóból és megadja, hogy melyik nap melyik ételeket kell elfogyasztanod.

Bemenet

A standard bemenet első sora az elérhető ételek N darabszámát tartalmazza. Az ételeket az $1, 2, \dots, N$ számokkal azonosítjuk.

A következő N sorban soronként egy-egy étel adatai szerepelnek: az S_i szavatossági ideje és az M_i minősége.

Kimenet

A standard kimenet első sorába azoknak a napoknak a K darabszáma kerüljön, ahány napig megoldható a napi étkezés. A következő K sor közül az i -edik sorban az i -edik napon elfogyasztott ételek azonosítói szerepeljenek.

Több megoldás esetén bármelyik megadható.

Példa

| Bemenet | Kimenet |
|---------|---------|
| 6 | 2 |
| 1 4 | 3 1 |
| 1 5 | 5 4 |
| 2 5 | |
| 6 9 | |
| 10 10 | |
| 10 1 | |

Korlátok

$$1 \leq N \leq 2000$$

$$1 \leq S_i \leq 2000 \text{ minden } i = 1 \dots N\text{-re}$$

$$1 \leq M_i \leq 10 \text{ minden } i = 1 \dots N\text{-re}$$

Időlimit: 1.0 mp.

Memórialimit: 256 MB

Pontozás

A megoldásodat több különböző tesztesetre lefuttatjuk. A teszteseteknek önálló pontértéke van és a megoldás pontszáma a megoldott tesztesetek pontszámainak összege. A feladat összpontszáma a legtöbb pontot elért megoldás pontszáma.

A pontszám 13%-a szerezhető olyan tesztekre, ahol $N \leq 8$.

A pontszám további 27%-a szerezhető olyan tesztekre, ahol $M_i \geq 5$ minden $i = 1 \dots N$ -re.

A pontszám további 20%-a szerezhető olyan tesztekre, ahol $S_i = N$ minden $i = 1 \dots N$ -re.

A pontszám további 20%-a szerezhető olyan tesztekre, ahol $N \leq 500$.

Minden **tesztesetben** részpontokat lehet szerezni. A teszteset pontszámának 50%-a jár abban az esetben, ha K értéke helyes, de a kimenet hibás konstrukciót tartalmaz, vagy csak a K értékből áll. Ha a programod kimenete tartalmaz konstrukciót, akkor ügyelj arra, hogy K darab sorban soronként két egész értéket tartalmazzon, különben az értékelő formai hiba miatt nem biztos, hogy megadja a részpontokat.