

Legcsalódottabb versenyző

Egy programozási verseny végén a versenyzők nevezéskor kapott sorszáma szerinti sorrendben közlik az elért pontszámokat. A legcsalódottabb az a versenyző, aki a legtovább hihette, hogy ő nyerte meg a versenyt, de később mégis megelőzte valaki (azaz nála magasabb pontszámot ért el).

Készíts programot, amely megadja a legcsalódottabb versenyzőt!

Bemenet

A *standard bemenet* első sorában a versenyzők száma van ($1 \leq N \leq 100\,000$).

A második sorban a versenyzők pontszámai találhatók ($1 \leq P_i \leq 100\,000$), a nevezéskor kapott sorszámuik szerinti sorrendben.

Kimenet

A *standard kimenet* első sorába a legcsalódottabb versenyző sorszámát kell kiírni! Ha több versenyző is azonos ideig hihette azt, hogy ő nyerte meg a versenyt, akkor közülük a legkisebb sorszámút kell megadni.

Ha nincs megoldás, azaz nem létezik legcsalódottabb versenyző, akkor a program -1-et írjon ki!

Példa

Bemenet

10

2 5 3 8 2 5 4 9 8 9

Kimenet

4

Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

Vásárlások inflációs hatása

Az Emezen Rt. online áruház furcsa módon változtatja a termékek árát. Segíts nekik kiszámolni, hogy a készletük eladásából legalább mekkora bevételük lesz!

Az áruházban N -féle termékből 1-1 darab van. Az i . termék kezdetben P_i bájt dollárba kerül. Az áruház költségei az infláció miatt folyamatosan nőnek, így a vezetőség egy új módszert vezetett be az árazásra: ha eladnak egy terméket x bájt dollárért, akkor K darab másik termék árát x bájt dollárral automatikusan megnövelik (ha már nincs K darab másik termék az áruházban, akkor minden ott levő termék árát növelik).

A vezetőség tudni szeretné, hogy a jelenlegi árukészletet a fenti árazási módszerrel legalább mennyi bájt dollárért tudja értékesíteni.

Tudjuk, hogy minden terméket el fognak adni és amíg ez meg nem történik, nem érkezik újabb árukészlet. Az eladások sorrendjét és hogy az egyes eladások után mely termékek ára növekszik nem ismerjük, ezek tetszőleges (a cég által nem megválasztott) módon történhetnek.

Írj programot, ami meghatározza az áruk eladásából a fenti módszerrel számolt lehető legkisebb bevételt!

Bemenet

A *standard bemenet* első sorában a termékek száma ($1 \leq N \leq 1000$) és az eladásokkor növekedő árú termékek darabszáma ($1 \leq K \leq 10$) található.

A második sorban N darab egész szám van, az áruházban levő termékek kiindulási árai ($1 \leq P_i \leq 1000$).

Kimenet

A *standard kimenetre* egyetlen egész szám kerüljön, a lehető legkisebb bevétel értéke!

Példa

Bemenet	Kimenet
4 3	22
1 3 1 4	

Korlátok

Időlimit: 0.2 mp.

Memórialimit: 64 MB

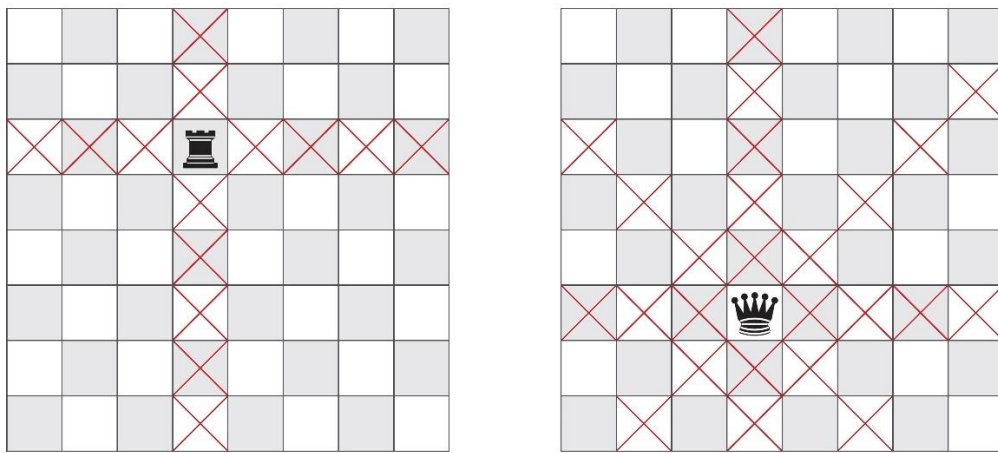
Ütős helyzet

Pali csak pár napja hallott a világ egyik legnépszerűbb játékaról, a sakkról. Most tanulja a bábuk lehetséges lépéseit.

Azt már tudja, hogy

- a **bástya** vízszintesen vagy függőlegesen léphet, míg egy másik bábu az útjába nem kerül, azaz a saját sorában, a saját oszlopában tud ütni.
- a **királynő** egyenesen vagy átlósan tetszőleges számú mezőt léphet, míg egy másik bábu az útjába nem kerül, mert azt leütheti, azaz a saját sorában, a saját oszlopában és az általa meghatározott két átlós irányban tud ütni.

Az ábrán látható, hogy egy bástya, illetve egy királynő mely mezőkön levő bábukat üti:



Pali N lépésben, egyesével próbál lehelyezni bábukat a 8×8 sakktáblára, azonban csak akkor teszi ezt meg, ha az épp lehelyezni kívánt bábút semelyik másik nem üti és a lehelyezni kívánt bábu sem üt más bábút.

Egy-egy lehelyezést egy $T \ X \ Y$ számhármassal ír le, ahol a T a lehelyezés típusa és az (X, Y) a hely koordinátái (X a sor száma, Y az oszlop sorszáma, a bal felső saroktól számozva):

- $T=1$ esetén Pali egy **bástyát** szeretne letenni az X . sorban az Y . helyre;
- $T=2$ esetén Pali egy **királynőt** szeretne letenni az X . sorban az Y . helyre;
- $T=3$ esetén az az X . sorban az Y . helyen álló bábút **leveszi** a tábláról (és az általa „ütésben levő” mezők felszabadulnak). Ebben az esetben biztosan tudjuk, hogy az (X, Y) helyen állt valamilyen bábu.

Lehelyezésenként a kimenetre

- az „**IGEN**” szöveget kell írni, ha a bábút Pali le tudja helyezni a megadott helyre (ezt meg is teszi);
- a „**NEM**” szöveget, ha így lenne két egymást ütő bábu (ebben az esetben a bábu nem kerül a táblára)
- $T=3$ esetén a sorban egy „-” (gondolatjel) szerepeljen!

Készíts programot, amely megmondja minden próbálkozásra, hogy lehelyezheti-e a bábút az adott helyre!

Bemenet

A *standard bemenet* első sorában a lerakások száma van ($1 \leq N \leq 1000$).

A következő N sorban egy lehelyezés adatai, a T , X és Y egész számok találhatóak, ahol T a lerakás típusa ($1 \leq T \leq 3$) és az $X Y$ a lerakni kívánt bábu táblán elfoglalt helye ($1 \leq X, Y \leq 8$, a számozást a tábla bal felső sarkától kezdjük).

Kimenet

A *standard kimenet* N darab sorának mindegyikébe, soronként a fenti leírásnak megfelelően az „IGEN”, a „NEM” vagy a „-” szöveg kerüljön!

Példa

Bemenet	Kimenet
5	IGEN
1 1 1	NEM
2 2 2	IGEN
1 2 2	-
3 2 2	-
3 1 1	

Példa

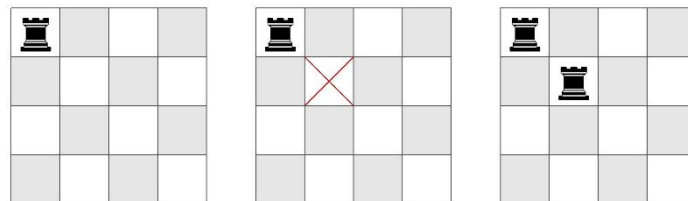
A képeken egy sakktabla bal felső részén sorban a lépések, a kiírt eredmény és a tábla állása:

Az ábra első sora:

1 1 1 IGEN

2 2 2 NEM

1 2 2 IGEN

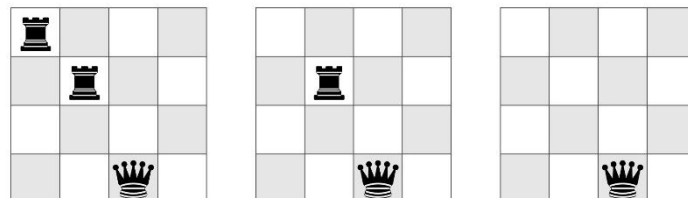


Az ábra második sora:

2 4 3 IGEN

3 1 1 -

3 2 2 -

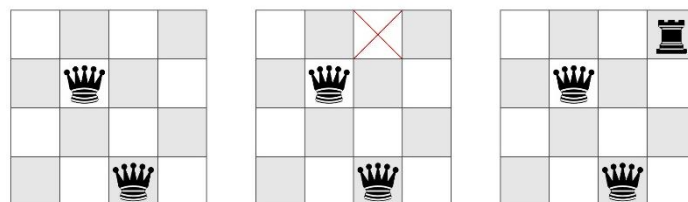


Az ábra harmadik sora:

2 2 2 IGEN

1 1 3 NEM

1 1 4 IGEN



Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

Pontozás

A pontszám 20%-a szerezhető olyan tesztekre, ahol minden $T=1$.

Zebra

Carnada Óttowal városában járunk. Itt az autósok kényelme érdekében a gyalogosan közlekedők jogai eléggé csorbultak, például új szabály, hogy egy zebrán gyalogosan csak úgy szabad átkelni, ha egyszerre mindkét irányban áthalad legalább egy-egy gyalogos.

Egy adott zebra egy út két oldalát köti össze. A zebrához a feladatunkban N gyalogos érkezik bizonyos időpontokban. Mindegyik gyalogosról tudjuk, hogy melyik oldalra akar eljutni és hogy mikor érkezik meg a zebrához. Az átkelés ideje elhanyagolható.

Kérdés, hogy ha a gyalogosok megfelelően dolgoznak össze, akkor mennyit kell legalább várakozniuk összesen (azaz mennyi a várakozásaik összegének a minimuma)?

Ha egy gyalogos áthaladt az úton, akkor rögtön elhagyja a zebra környékét, egyébként amíg nem kel át, addig akármilyen hosszú ideig is várakozhat. Ha mindkét oldalon várakozik egy adott időpontban gyalogos, akkor is dönthetnek úgy, hogy tovább várakoznak.

Írj programot, ami meghatározza a minimális összes várakozás időtartamát!

Bemenet

A *standard bemenet* első sorában a gyalogosok száma ($1 \leq N \leq 80$) található.

A következő sorban N darab egész szám található: az i . szám az i . gyalogos átkelési iránya ($c_i=0$ vagy $c_i=1$, attól függően, hogy a gyalogos melyik oldalról melyik oldalra akar átjutni).

A harmadik sorban N darab egész szám található, az i . értéke az i . gyalogos érkezésének ideje a zebrához ($1 \leq t_i \leq 10^6$).

Kimenet

A *standard kimenetre* egyetlen szám kerüljön, a lehető legrövidebb összes várakozási idő!

Példa

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

Korlátok

Időlimit: 0.7 mp.

Memórialimit: 64 MB