Fénysor

Egy karácsonyi fénysor N izzóból áll. Az izzókat 1-től N-ig sorszámozzuk. Sajnos a fénysor hibás: jelenleg bizonyos izzók világítanak rajta, más izzók viszont ki vannak kapcsolva.

Egy lépésben a következő módosítást tudjuk végrehajtani a fénysoron: Kiválasztunk egy izzót, ami nem a fénysor szélén van (tehát a 2, 3, ...N − 1 sorszámú izzók valamelyikét), és átkapcsoljuk. Ekkor azonban az izzó bal- és jobboldali szomszédja is állapotot vált (ha égett, akkor kikapcsol, ha nem égett, akkor bekapcsol).

Jelöljük egy izzó kikapcsolt állapotát a 0, bekapcsolt állapotát pedig az 1 számjeggyel. Például, ha egy N = 5 izzóból álló fénysor állapota jelenleg [0, 0, 1, 1, 0], akkor a harmadik izzó átkapcsolása után az állapota [0, 1, 0, 0, 0]-ra változna, míg a negyedik izzó átkapcsolása után a [0, 0, 0, 0, 1] állapothoz jutnánk.

Írj programot, ami megad egy olyan lépéssorozatot (ha létezik), aminek végrehajtása után az összes izzó világít.

Bemenet

A standard bemenet első sorában az izzók N száma található.

A második sor N darab egész értéket tartalmaz, a fénysor kezdeti állapotát leíró A_i értékeket.

Kimenet

A standard kimenet első és egyetlen sorába a −1 érték kerüljön, ha nem lehet elérni, hogy az összes izzó világítson.

Minden egyéb esetben az első sorba az összes izzó felkapcsolásához használt lépések K száma kerüljön. A második sorba K pozitív egészet kell kiírni, azoknak az izzóknak a sorszámait, amiket átkapcsolunk (az átkapcsolásuk sorrendjében).

Legfeljebb 2 · N lépés használata engedélyezett. Belátható, hogy ha megoldható a feladat, akkor ennyi lépés biztosan elegendő. A lépések számát nem szükséges minimalizálni. Több helyes megoldás létezése esetén bármelyik megadható.

Példa

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

A fénysor állapota az egyes lépéseket követően:

- Harmadik izzó átkapcsolása: [0, 0, 1, 0, 0]
- Második izzó átkapcsolása: [1, 1, 0, 0, 0]
- Negyedik izzó átkapcsolása: [1, 1, 1, 1, 1]

Bemenet	Kimenet
5 0 0 1 1 0	−1

Bemenet	Kimenet
5 1 1 1 1 1	0

Korlátok

$3 \leq N \leq 50\,000$

$0 \leq A_i \leq 1$ minden $i = 1 \ldots N$-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 12%-a szerezhető olyan tesztekre, ahol $N = 3$.

A pontszám további 16%-a szerezhető olyan tesztekre, ahol pontosan egy lépéssel teljesíthető a feladat.

A pontszám további 16%-a szerezhető olyan tesztekre, ahol kezdetben egyik izzó sem világít.

A pontszám további 28%-a szerezhető olyan tesztekre, ahol $N \leq 12$.

Munkák

Egy vállalkozónak N munkát kell elvégeznie, amiket az 1, 2, ..., N számokkal azonosítunk. Minden munkát egy nap alatt lehet teljesíteni és a vállalkozó egy nap csak egy munkát tud végezni.

Minden munkához szükség lehet hiányzó alapanyagok beszerzésére, ezért minden munkának van egy **kezdő határideje**. Az i sorszámú munka kezdő határideje H_i, ami azt jelenti, hogy a munkát legkorábban a H_i-edik napon lehet elvégezni. A napokat 1-től kezdve számozzuk.

A vállalkozó szeretne a lehető legkorábban végezni a munkákkal. Írj programot, amely megadja a munkáknak egy olyan beosztását, aminél a vállalkozó a lehető legkevesebb nap alatt végez.

Bemenet

A `standard bemenet` első sora a munkák N számát és a munkák kezdő határidejének legnagyobb K értékét tartalmazza. A második sor pontosan N darab egész számot tartalmaz, rendre a munkák H_i kezdő határidejét.

Kimenet

A `standard kimenet` első sorába azt a legkisebb M egész számot kell írni, amelyre teljesül, hogy minden munkát el lehet végezni az M-edik napig. A második sor pontosan N egész számot tartalmazzon, az i-edik szám annak a napnak a sorszáma legyen, amelyik napon az i-edik munkát végzi a vállalkozó. Több helyes megoldás esetén bármelyik megadható.

Példa

Bemenet	Kimenet
7 10	9
4 3 6 3 7 3 7	6 3 7 4 8 5 9

Korlátok

$5 \leq N \leq 100\,000$

$1 \leq K \leq 100\,000$

$1 \leq H_i \leq K$ minden $i = 1 \ldots N$-re

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 13%-a szerezhető olyan tesztekre, ahol $N \leq 100$ és $K \leq 2000$, továbbá a munkák H_i kezdő határidejei különbözők.

A pontszám további 14%-a szerezhető olyan tesztekre, ahol $N \leq 100$.

A pontszám további 20%-a szerezhető olyan tesztekre, ahol $N \leq 20\,000$.

Minden tesztesetre **részpontokat** lehet szerezni: a pontszám 40%-a jár, ha az első sorba kiírt szám (az utolsó munkanap lehető legkisebb értéke) helyes.

Cukorkák

Egy boltban `N` darab cukorka kapható. Az `i`-edik cukorka édességét egy A_i pozitív egész értékkel jellemezzük.

Sári szeretné megvásárolni a cukorkák egy részét, majd a megvett cukorkákat elosztani a barátnője és saját maga között úgy, hogy a kapott cukorkákkal mindketten **elégedettek** legyenek. Mindkettejükre teljesül, hogy pontosan akkor elégedettek a megkapott cukorkákkal, ha

- legalább egy cukorkát kapnak, és
- bármely két kapott cukorka édessége legfeljebb `K`-val tér el egymástól.

Például ha valaki pontosan egy cukorkát kap, akkor ő elégedett lesz.

Írj programot, ami kiszámítja, hogy összesen legfeljebb hány cukorkát tud megvásárolni és szétosztani Sári úgy, hogy mindketten elégedettek legyenek a kapott cukorkákkal.

Bemenet

A `standard bemenet` első sorában a cukorkák `N` száma és a `K` érték található.

A második sor `N` pozitív egész számot tartalmaz, a cukorkák $A_1, A_2, \ldots, A_N$ édességeit.

Kimenet

A `standard kimenetre` egyetlen egész számot kell írni, a megvásárolható cukorkák maximális számát.

Példa

Bemenet

```
6 3
1 5 3 8 2 7
```

Kimenet

```
6
```

Sári az `1, 2, 3`, a barátnője az `5, 7, 8` édességű cukorkákat kapja. Mindkét csoporton belül bármley két édesség eltérése legfeljebb `3`.

Bemenet

```
5 1
3 1 4 1 5
```

Kimenet

```
4
```

Egy optimális vásárlás és elosztás: egyiküknek az `1, 1` és a másikuknak a `4, 5` édességű cukorkák.

Korlátok

$2 \le N \le 200\,000$

$0 \le K \le 10^9$

$1 \le A_i \le 10^9$ minden $i = 1 \ldots N$-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 20%-a szerezhető olyan tesztekre, ahol $N \leq 10$.

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

A pontszám további 20%-a szerezhető olyan tesztekre, ahol $K = 0$.

Három kupac

Anna és Bob unatkoznak, ezért kitaláltak egy egyszerű játékot, hogy szórakoztassák magukat. A játék kezdetén kavicsokat helyeznek az asztalra `3` kupacba. Az első kupacban `x`, a másodikban `y` és a harmadikban `z` kavics van. Legalább egy kavics van az asztalon, de előfordulhat, hogy valamelyik kupacban nincsen kavics.

A játékban Anna és Bob felváltva lépnek. Anna kezdi a játékot. Mindkettejüknek a saját lépésében ki kell választania két olyan kupacot, amiben van legalább egy kavics, és mindkét kupacból el kell vennie egy-egy kavicsot. Ezután a harmadik kupachoz hozzá kell tenni egy új kavicsot.

A játék addig folytatódik, amíg valamelyik játékos nem tud már lépni. Ez a játékos elveszíti a játékot. Írj programot, ami `T` lehetséges kezdőállásra megadja, hogy melyikük fogja nyerni a játékot, ha mindketten optimálisan játszanak.

Bemenet

A `standard bemenet` első sorában a kezdőállások `T` száma található.

A következő `T` sor mindegyike egy kezdőállást megadó számhármast tartalmaz: rendre az `x`, `y` és `z` egész értékeket.

Kimenet

A `standard kimenetre` pontosan `T` sort kell kiírni, minden kezdőállásra egyet. Az `i`-edik sorba annak a játékosnak a nevét kell kiírni (‘`Anna`’ vagy ‘`Bob`’), aki meg tudja nyerni a játékot függetlenül attól, hogy hogyan játszik az ellenfele.

Példa

Bemenet	Kimenet
`2`	`Anna`
`3 0 1`	`Bob`
`2 2 1`	

Az első játékban a kavicsok száma a kupacokban kezdetben `(3, 0, 1)`.

- Annának egyetlen lehetséges lépése van: az első és a harmadik kupacból elvesz egy kavicsot, a másodikhoz hozzáad egyet. Ezt követően a kavicsok száma: `(2, 1, 0)`.
- Most Bob egyetlen lehetséges lépése, hogy az első és a második kupacból vesz el, a harmadikhoz hozzáad egy kavicsot. A kavicsok száma a lépés után: `(1, 0, 1)`.
- Annanak muszáj az első és harmadik kupacból elvennie és a másodikhoz hozzátennie egy kavicsot. A lépése után a kavicsok száma: `(0, 1, 0)`.
- Bobnak már nincs szabályos lépése, így elveszíti a játékot.

A második játékban Annának kétféle különböző kezdőlépése lehet:

- Az első két kupacból vesz el, aminek következtében `(1, 1, 2)`-re módosul a kavicsok száma. Ha most Bob is az első két kupacból vesz el, akkor `(0, 0, 3)` kavics marad és Anna veszít.
- Az első két kupac egyikéből és a harmadik kupacból vesz el, így `(3, 1, 0)` vagy `(1, 3, 0)` kavics lesz a lépés után a kupacokban. Az első játéknál már beláttuk, hogy ezekben a helyzetekben a soron következő játékos fog nyerni, aki ezúttal Bob lesz.

Tehát bármilyen lépéssel kezdi Anna a játékot, Bob megfelelő stratégiával végül nyerni tud.

Korlátok

$1 \leq \texttt{T} \leq 50$

$0 \leq \texttt{x, y, z} \leq 100\,000$

$1 \leq \texttt{x} + \texttt{y} + \texttt{z}$

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 8%-a szerezhető olyan tesztekre, ahol $\texttt{x} = 0$ és $\texttt{y} = 1$.

A pontszám további 8%-a szerezhető olyan tesztekre, ahol $\texttt{x, y, z} \leq 2$.

A pontszám további 12%-a szerezhető olyan tesztekre, ahol $\texttt{x, y, z} \leq 3$.

A pontszám további 40%-a szerezhető olyan tesztekre, ahol $\texttt{x, y, z} \leq 50$.