

Oszthatóság 31-gyel

31-gyel úgy vizsgálhatjuk meg az oszthatóságot, hogy a szám első számjegyétől az utolsó előtti számjegyéig képzett számból kivonjuk az utolsó számjegy háromszorosát. Ha ez a szám osztható 31-gyel, akkor az eredeti is. Pl.: $204197 \rightarrow 20419 - (3 \cdot 7) = 20398 \rightarrow 2039 - (3 \cdot 8) = 2015 \rightarrow 201 - (3 \cdot 5) = 186 \rightarrow 18 - (3 \cdot 6) = 0$. 0 osztható 31-gyel (mert 0 minden számmal osztható), ezért 204197 is osztható 31-gyel. Az ismétlés megáll, ha a kapott szám nullánál kisebb vagy egyenlő lenne.

Írj programot, amely eldönti egy számról, hogy osztható-e 31-gyel!

Bemenet

A *standard bemenet* első sorában a szám szerepel ($1 \leq N \leq 1\,000\,000\,000$), aminek a 31-gyel oszthatóságát vizsgáljuk.

Kimenet

A *standard kimenet* első sorába az IGEN vagy a NEM szót kell írni, attól függően, hogy N osztható-e 31-gyel! A második sorba a fenti módszerrel kiszámolt közbülső számok kerüljenek, a kiszámítás sorrendjében! Ha az utolsó szám 0, azt még ki kell írni! Üres sort kell kiírni, ha nincs egyetlen közbülső szám sem!

Példa

| | |
|---------|-------------------------------|
| Bemenet | Kimenet |
| 204197 | IGEN 20398 2015 186 0 |
| Bemenet | Kimenet |
| 204196 | NEM 20401 2037 182 12 |
| Bemenet | Kimenet |
| 62 | IGEN 0 |
| Bemenet | Kimenet |
| 63 | NEM {üres második sor} |

Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

Inverzió

Egy számsorozatban az $1 \dots n$ számok szerepelnek valamilyen sorrendben, mindegyik pontosan egyszer. Inverziónak nevezzük az olyan $i < j$ indexeket, ahol az i . szám nagyobb a j .-nél. Ekkor az inverzió távolságán a $j - i$ különbséget értjük.

Készíts programot, amely megadja a számsorozat egy maximális távolságú inverzióját!

Bemenet

A *standard bemenet* első sorában a sorozat hossza van ($1 \leq N \leq 500\,000$). A második sorban a sorozat elemei találhatóak ($1 \leq S_i \leq N$, különbözőek)

Kimenet

A *standard kimenet* első sorába -1 kerüljön ha a sorozat nem tartalmaz inverziót, egyébként a maximális távolságú inverzió indexeit kell kiírni, több lehetséges esetén tetszőlegesen!

Példa

| Bemenet | Kimenet |
|-----------|---------|
| 5 | 1 5 |
| 4 3 5 1 2 | |

Korlátok

Időlimit: 0.5 mp.

Memórialimit: 64 MiB

Pontozás

A pontok 20%-a kapható olyan tesztekre, ahol $N \leq 1000$.

Késés

A 12.A bizonyos tanulóinak birtokába került a jövő hét keddi 8:00-s történelem óra tananyaga. Ez egy N hosszú számsorozat, amelynek i . eleme az óra i . pillanatában elhangzó évszám. A tanulók furfangosak lévén észrevették hogy az évszámok ismétlődhetnek.

Készíts programot, amely megadja azt a legkésőbbi időpillanatot, amikor megérkezve még az összes órán elhangzó évszámról hallanak legalább egyszer!

Bemenet

A *standard bemenet* első sorában az óra időpillanatainak száma található ($1 \leq N \leq 200\,000$). A második sor i : száma az i . pillanatban elhangzó évszám ($1 \leq E_i \leq 10^9$).

Kimenet

A *standard kimenet* első sorába azt a legkésőbbi időpillanatot kell írni, amelyiktől kezdve az órát ugyanazokról az évszámokról hallanak, mintha pontosan kezdték volna (az 1. időpillanatban) az órát!

Példa

| Bemenet | Kimenet |
|------------------------------|---------|
| 6 | 3 |
| 1848 1849 1848 1849 895 1516 | |

Korlátok

Időlimit: 0.5 mp.

Memórialimit: 64 MiB

Pontozás

A pontok 20%-a szerezhető olyan tesztekre, ahol $N \leq 1000$ és $E_i \leq N$.

A pontok további 60%-a szerezhető olyan tesztekre, ahol minden $E_i \leq N$.

Színezés

Van N darab piros és kék golyó egymás mellett egy sorban elrendezve. Szeretnénk, ha az összes ugyanolyan színű lenne (mindegy, hogy piros vagy kék). Ehhez színező műveleteket hajthatunk végre. Egy színező műveletben vehetünk akármennyi egymás melletti ugyanolyan színű golyót és átszínezzük őket a másik színre. A golyók színei folyamatosan változnak, összesen Q változás történik. Az i . változásban egy adott golyó az ellenkező színűre változik és úgy is marad a későbbi műveletek szempontjából.

Írj programot, amely megadja, hogy minimálisan hány ilyen művelettel tudjuk elérni azt hogy minden golyó ugyanolyan színű legyen! Add meg ezt az 1. változás előtt, illetve mindegyik változás után

Bemenet

A *standard bemenet* első sorában a golyók száma ($1 \leq N \leq 200\,000$) és a változások száma ($1 \leq Q \leq 200\,000$) szerepel. A második sorban N karakter van, 0 jelöli a piros, 1 pedig a kék golyót. A következő Q sor közül az i -ben az i . változtatás során megváltozott színű golyó sorszáma van ($1 \leq K_i \leq N$).

Kimenet

A *standard kimenet* első sorába a változtatások előtti szükséges színező műveletek számát kell írni, a következő Q sorba pedig az egyes változtatások után szükséges színező műveletek számát!

Példa

| Bemenet | Kimenet |
|---------|---------|
| 5 3 | 2 |
| 01001 | 1 |
| 2 | 1 |
| 1 | 1 |
| 5 | |

Korlátok

Időlimit: 0.6 mp.

Memórialimit: 64 MiB

Pontozás

A pontok 20%-a szerezhető olyan tesztekre, ahol $N, Q \leq 15$.

A pontok további 20%-a szerezhető olyan tesztekre, ahol $N * Q \leq 1\,000\,000$.

Toronyépítés (1,1,3,3)

Négyféle elemünk (építőköcánk) van, mindegyikből tetszőleges számú. A piros és a zöld elemek magassága egy, a fehéré és a sárgáé három.

A mintán alul egy zöld, felül egy piros kocka van, középen pedig egy fehér tégl.

Írj programot, amely megadja, hogy hány különböző N magasságú torony építhető belőlük! Mivel ez a szám nagyon nagy is lehet, az eredménynek a 20210108-cal vett osztási maradékát kell megadni!

Bemenet

A *standard bemenet* első sorában a torony magassága szerepel ($1 \leq N \leq 1\,000\,000$).

Kimenet

A *standard kimenet* első sorába az építhető N magasságú, különböző tornyok számának 20210108-cal vett osztási maradékát kell írni!

Példa

| | |
|---------|----------|
| Bemenet | Kimenet |
| 5 | 56 |
| Bemenet | Kimenet |
| 2021 | 20016480 |

Korlátok

Időlimit: 0.2 mp.

Memórialimit: 64 MB

Pontozás

A pontok 50%-a szerzhető olyan tesztekre, ahol $N \leq 50$.



Zenehallgatás

Petya összeállított egy lejátszási listát az N kedvenc dalából. Mindegyik dalnak ismerjük a hosszát másodpercben. A lejátszási listát végtelenítve hallgatja, tehát amint véget ért, előlről kezdi, és újra ugyanolyan sorrendben következnek a dalok. Felírt magának K időpontot, és kíváncsi arra, hogy ezekben az időpontokban melyik dal fog szólni.

Készíts programot, amely megadja, hogy az egyes időpontokban hányadik dal fog szólni!

Bemenet

A *standard bemenet* első sorában dalok száma ($1 \leq N \leq 100\,000$) és az időpontok száma ($1 \leq K \leq 100\,000$) van. A második sor i . száma az i . dal hossza másodpercben ($1 \leq T_i \leq 10\,000$). A harmadik sorban a K időpont van ($1 \leq P_i \leq 10^9$), amelyekre meg kell határozni, hogy abban az időpontban melyik dal szól.

Kimenet

A *standard kimenet* első sorába egyetlen sora K dal sorszámát tartalmazza, ahol az i -edik szám a P_i -edik másodpercben szóló dal sorszámát.

Példa

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

Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

Pontozás

A pontok 30%-a szerezhető olyan tesztekre, ahol $K \leq 1000$ és $N \leq 1000$.