

Problema Joker

Fișier de intrare `stdin`
Fișier de ieșire `stdout`



Abilitățile descrise mai jos ale acestui Joker nu sunt canon.

Te joci un joc cu *Joker*-ul. El ți-a dat un număr N ($5 \leq N \leq 1000$), și ți-a spus că are ascunse două numere L și R ($0 \leq L \leq R \leq N - 1$). Scopul jocului este să găsești acestea. Ați stabilit că, pentru a îndeplini acest scop, tu poți să îi pui întrebări de următoarea formă:

Îți dau un șir de N numere a_0, a_1, \dots, a_{N-1} . Spune-mi care este minimul numerelor a_L, a_{L+1}, \dots, a_R .

Cu toate acestea, a intervenit o problemă: În limba nativă a *Joker*-ului, nu există un cuvânt pentru minim, mai degrabă, el a înțeles cuvântul *extrem*. Astfel, el va răspunde fiecărei întrebări cu **maximul** sau **minimul** segmentului $[L, R]$ din șirul pe care i l-ați dat. Mai mult decât atât, este posibil să îți răspundă cu maximul sau minimul doar ca să te încurce și mai tare.

Joker-ul știe că acest joc este probabil prea greu de câștigat cu forțe proprii. De aceea, el consideră orice pereche de numere (L', R') un răspuns *corect*, o pereche care respectă:

$$|L - L'| + |R - R'| \leq 1, \text{ unde s-a notat cu } |a| \text{ valoarea absolută a numărului } a.$$

Cerință

În lumina acestor informații, trebuie să faci tot ce este posibil să găsești un răspuns care să satisfacă criteriul de corectitudine al *Joker*-ului cu număr minim de întrebări.

Interacțiune

Participanții au la dispoziție următoarele funcții pentru a interacționa cu *Joker*-ul:

- Funcția

```
int query(std::vector<int> a);
```

Această funcție primește un șir de numere a_0, a_1, \dots, a_{N-1} și returnează fie **minimul** sau **maximul** pe segmentul $[L, R]$ al șirului dat, ales arbitrar de către **Joker**-ul.

- Funcția

```
void answer(int L, int R);
```

Participanții trebuie să apeleze această funcție pentru a furniza răspunsul final, cu perechea (L, R) presupus corectă, pe care au dedus-o.

Participanții trebuie să implementeze doar funcția `solve`, care este apelată pentru a rezolva jocul. Nu trebuie să implementeze funcția `main`. Codul sursă trebuie să înceapă cu

```
#include "joker.h"
```

și să implementeze doar funcția `solve(int N);`.

Important: Tot codul participantului ar trebui să fie scris în `solutie.cpp`. Acolo participanții pot defini variabile globale și funcții proprii pentru a facilita rezolvarea problemei.

Fișiere Atașate

Participanții au acces la următoarele fișiere:

- `joker.h` - header-ul care definește funcțiile de interacțiune (`query` și `answer`) necesare pentru implementare.
- `sample_submission.cpp` - un exemplu simplu de implementare care execută un `query` și răspunde cu $L = R = 0$ de fiecare dată.
- `Lgrader.cpp` - un grader de testare pentru participanți, nu cel utilizat oficial, care îi poate ajuta să verifice codul.

Utilizarea Lgrader în Code::Blocks

Pentru a utiliza `Lgrader.cpp` în Code::Blocks:

1. Creați un proiect nou.
2. Adăugați un fișier denumit `solutie.cpp` (selectați `File > New > Empty File`), unde veți scrie codul soluției.
3. Adăugați un fișier gol numit `joker.h` și copiați conținutul din fișierul atașat `joker.h`.
4. Copiați conținutul din `Lgrader.cpp` în `main.cpp` pentru a simula rularea soluției.

După acest setup, scrieți în `solutie.cpp` implementarea funcției `solve`. Pentru a testa configurația, puteți copia inițial codul din `sample_submission.cpp` în `solutie.cpp`. Apoi, apăsați `F9` sau selectați **Build and Run** pentru a compila și rula programul.

La rulare, programul va citi valorile N, L și R , și va simula interacțiunea dintre `Lgrader` și soluția concurentului, afișând răspunsurile la interogările făcute prin `query` și validând răspunsul final oferit prin `answer`.

Exemplu de Interacțiune

Considerăm $N = 5$ și valorile ascunse $L = 1, R = 3$. Segmentul $[L, R] = [1, 3]$ corespunde subșirului $\{3, 8, 1\}$ din vectorul de intrare. Grader-ul va apela funcția `solve(5)`.

Din interiorul funcției `solve`, dacă participantul apelează:

```
int query({5, 3, 8, 1, 6});
```

Output-ul va fi fie 1 (**minimul**) sau 8 (**maximul**) pe segmentul $[L, R] = [1, 3]$, ales arbitrar de către **Joker**-ul.

Dacă participantul apelează:

```
int query({2, 4, 7, 5, 9});
```

Output-ul va fi fie 4 (**minimul**) sau 7 (**maximul**) pe segmentul $[L, R] = [1, 3]$, ales arbitrar.

Din interiorul aceleiași funcții `solve`, dacă participantul apelează:

```
answer(1, 3);
```

Output-ul va fi:

```
Correct
```

Participantul va primi un punctaj calculat conform formulei de evaluare de la secțiunea **Grading**.

Grading

Fie $Q_{participant}$ numărul de query-uri folosite de soluția voastră și Q_{optim} numărul minim de query-uri.

Fie $x = \frac{Q_{participant} - Q_{optim}}{Q_{optim}}$.

Atenție!, nu puteți face mai mult de 10 000 de query-uri per test.

Punctajul primit pe un test este

$$0.2 + \frac{0.2}{e^{2x} + 1} + \frac{0.7}{1 + x}$$