

## Monede

100 puncte

Se consideră un șir de  $N$  monede numerotate de la  $0$  la  $N-1$ . La o aruncare, moneda  $i$  are probabilitatea  $p[i]$  să cadă pajură, și  $1 - p[i]$  să cadă cap.

### Cerința

Pentru  $Q$  intervale  $[a, b]$  se cere să se afle probabilitatea ca, dacă s-ar arunca cu monedele  $a, a+1, \dots, b$ , un număr impar de monede să pice pajură.

### Interacțiune

Concurentul va include fișierul `monede.h`, realizând acest lucru prin directiva de preprocesor `#include "monede.h"`. În acest header este implementată o structură de date `Raport`. Aceasta se poate utiliza în felul următor.

```
Raport r; // r se initializeaza cu valoarea 0 / 1 = 0
Raport p(1, 2); // p reprezinta raportul 1 / 2
Raport q = Raport(2, 3); // q reprezinta raportul 2 / 3
bool b = (p == q); // b este fals, caci 1 / 2 != 2 / 3
Raport r = p + q; // r reprezinta 1 / 2 + 2 / 3 = 7 / 6.
Raport r = p - q; // r reprezinta 1 / 2 - 2 / 3 = - 1 / 6.
Raport r = p * q; // r reprezinta 1 / 2 * 2 / 3 = 1 / 3.
Raport r = p / q; // r reprezinta 1 / 2 : 2 / 3 = 3 / 4.
// Se poate calcula si p * q, p / q, p - q
r -= p; // r devine 7 / 6 - 1 / 2 = 2 / 3.
// Se putea efectua si r += p, r *= p, r /= p.
```

Concurentul va implementa două funcții. Prima este funcția `init`, având semnătura:

```
void init(int N, const Raport p[]);
```

Comisia va apela această funcție exact o dată, la începutul rulării programului. Funcția va primi prin parametrul  $N$  (numărul natural) numărul de monede și prin parametrul  $p$  șirul de probabilități, indexat de la  $0$ .

A doua funcție este `solve`, cu semnătura:

```
Raport solve(int a, int b);
```

Comisia va apela această funcție de  $Q$  ori. Funcția primește prin intermediul parametrilor  $a$  și  $b$  ( $0 \leq a \leq b < N$ ) două poziții. Funcția va trebui să returneze răspunsul pentru intervalul  $[a, b]$ .

Fișierul `skeleton.cpp`, ce îl găsiți în atașamente, vă arată structura programului ce trebuie să îl scrieți.

**Concurentul nu implementează funcția `main`, ci funcțiile indicate în enunț, și oricare funcții auxiliare de care are nevoie!**

### Restricții

- $1 \leq N \leq 1.000.000$
- $1 \leq Q \leq 1.000.000$
- Valorile  $a$  și  $b$  trimise prin parametri funcției `solve` respectă condiția  $0 \leq a \leq b < N$
- Pentru 23 puncte,  $N \leq 15$ ,  $Q \leq 100$ .
- Pentru alte 10 puncte,  $N \leq 1.000$ ,  $Q \leq 1.000$ .
- Pentru alte 37 de puncte,  $N, Q \leq 200.000$ .

- Pentru alte 15 de puncte,  $p[i]$  diferit de  $1 / 2$ .

**Exemplu**

Intrare	Ieșire
$N = 3$ $p = [1/4, 1/3, 1/2]$	$Solve(0, 0) = 1 / 4$ ; $Solve(1, 1) = 1 / 3$ ; $Solve(0, 1) = 5 / 12$ $Solve(0, 2) = Solve(1, 2) = Solve(2, 2) = 1 / 2$

**Limită de timp:** 0.6 secunde

**Memorie totală:** 256 MB