

## Problem Nice Lines

C header: `nice_lines.c.h`  
C++ header: `nice_lines.h`

Roxette, prințesa pirat, a ajuns la insula secretă în arhipelagul Remeian. Acolo se zvonește că este îngropată o faimoasă comoară, liniile frumoase aurite.

Insula secretă este un pătrat, cu lungimea laturii de  $2 \times 10^{12}$  metri. Orice punct de pe insulă este descris folosind coordonate carteziane, cu  $(0, 0)$  la centru, și cele două axe paralele cu laturile.

Sunt  $N$  *linii frumoase aurite* îngropate pe insulă. A  $i$ -a linie, pentru  $0 \leq i < N$ , ocupă mulțimea tuturor punctelor  $(x, y)$  cu valori reale descrise de ecuația liniară  $y = a_i x + b_i$ .

Roxette poate să folosească un aparat special, numit *lineometru*. Fiind dat orice punct  $p$  de pe insulă, *lineometrul* va calcula suma distanțelor<sup>1</sup> de la punctul  $p$  la fiecare dintre cele  $N$  *linii frumoase aurite*.

Din nefericire, *lineometrul* are un număr limitat de utilizări. O poți ajuta pe Roxette să identifice comoara cu un număr suficient de mic de utilizări ale *lineometrului*?

## Protocol de interacțiune

Concurentul trebuie să implementeze o funcție:

```
(C) void solve(int subtask_id, int N);  
(C++) void solve(int subtask_id, int N);
```

Această funcție va fi apelată **exact o dată**, la începutul interacțiunii.  $N$  este numărul de *linii frumoase aurite* ascunse pe insulă

Această funcție poate să apeleze altă funcție, dar **nu de mai mult de  $Q_{max}$  ori**:

```
(C) long double query(long double x, long double y);  
(C++) long double query(long double x, long double y);
```

Concurentul trebuie să apeleze această funcție doar cu argumente astfel încât  $-10^{12} \leq x, y \leq 10^{12}$ .

Aceasta returnează rezultatul *lineometrului* pentru un punct cu coordonatele carteziane  $(x, y)$  – i.e. suma distanțelor de la punctul  $(x, y)$  la fiecare dintre cele  $N$  *linii frumoase aurite*. Rețineți că *liniile frumoase aurite* nu vor fi furnizate, fiind obiectivul concurentului de a le găsi.

Când concurentul găsește cele  $N$  *linii frumoase aurite*, acesta trebuie să apeleze următoarea funcție:

```
(C) void the_lines_are(int* a, int* b);  
(C++) void the_lines_are(std::vector<int> a, std::vector<int> b);
```

Unde  $a[i]$  și  $b[i]$  trebuie să descrie cea de-a  $i$ -a *linie frumoasă aurită*, pentru  $0 \leq i < N$ . Concurentul poate să returneze liniile în orice ordine.

## Restricții

- $1 \leq N \leq 100$
- $-10\,000 \leq a_i, b_i \leq 10\,000$
- Oricare două linii nu sunt paralele.

<sup>1</sup>Distanța Euclidiană între un punct și o linie este lungimea celui mai scurt segment care atinge și linia și punctul.

## Punctaj

Pentru a calcula punctajul pentru un test procedați astfel:

- Fie  $Q$  numărul de apelări ale funcției `query`.
- Dacă  $Q > Q_{max}$ , sau dacă *liniile frumoase aurite* nu au fost corect raportate, atunci scorul pentru test va fi 0.
- Dacă  $Q \leq Q_{min}$ , atunci scorul pentru test va fi 1.
- Altfel, scorul pentru test va fi  $1 - 0.7 \cdot \frac{Q - Q_{min}}{Q_{max} - Q_{min}}$ .

Pentru a calcula scorul pentru un subtask, se ia scorul minim obținut pentru fiecare test din acel subtask și se înmulțește cu numărul total de puncte pentru acel subtask.

### Subtask 1 (11 puncte)

- $N = 1$
- $Q_{min} = 10\,000$ ,  $Q_{max} = 10\,000$

### Subtask 2 (13 puncte)

- $N = 2$
- $Q_{min} = 10\,000$ ,  $Q_{max} = 10\,000$

### Subtask 3 (7 puncte)

- $N = 3$
- $Q_{min} = 10\,000$ ,  $Q_{max} = 10\,000$

### Subtask 4 (19 puncte)

- $-500 \leq a_i, b_i \leq 500$
- $Q_{min} = 402$ ,  $Q_{max} = 10\,000$

### Subtask 5 (23 puncte)

- $N \leq 30$
- $Q_{min} = 402$ ,  $Q_{max} = 10\,000$

### Subtask 6 (27 puncte)

- $Q_{min} = 402$ ,  $Q_{max} = 10\,000$

## Exemplu

Apelurile comisiei	Apelurile concurentului
<pre>solve( /* subtask_id = */ 1, /* N          = */ 1)</pre>	<pre>query(0, 0) returns 0 query(1, 1) returns 0 the_lines_are( /* a = */ {1}, /* b = */ {0})</pre>