

Problema Nice Lines

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

La piratessa Roxette è arrivata nella misteriosa isola del tesoro nell'arcipelago Remeian, lì un famoso tesoro è sepolto. Le voci narrano che per trovare il tesoro è necessario seguire le *golden nice lines*!

L'isola è un quadrato grande $2 \cdot 10^{12} \times 2 \cdot 10^{12}$ metri, descritto da coordinate cartesiane dove $(0,0)$ è situato al centro dell'isola e gli assi sono paralleli ai lati del quadrato.

Le *golden nice lines* sono N rette nascoste nell'isola. L' i -esima retta ($0 \leq i < N$) occupa tutti i punti a coordinate reali (x,y) descritte dall'equazione $y = a_i x + b_i$.

Roxette può usare un *lineometro*, uno speciale strumento che, dato un punto p nell'isola, è in grado di calcolare la somma delle distanze¹ da p a tutte le N rette.

Sfortunatamente la batteria del *lineometro* è quasi scarica, quindi ha un numero limitato di utilizzi. Aiuta Roxette a trovare tutte le N *golden nice lines* prima che lo strumento si scarichi!

Implementazione

Devi implementare la seguente funzione:

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

Questa funzione verrà chiamata **esattamente una volta**, all'inizio dell'interazione. N è il numero di *golden nice lines* nascoste nell'isola.

La funzione può a sua volta chiamare la seguente funzione, ma **non più di Q_{max} volte**:

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

I parametri che passerai a questa funzione devono essere tali che $-10^{12} \leq x, y \leq 10^{12}$.

La funzione restituisce il risultato del *lineometro* applicato ad un punto con coordinate cartesiane (x,y) – ovvero, la somma delle distanze dal punto (x,y) a ciascuna delle N *golden nice lines*. Nota che le *golden nice lines* non vengono fornite come parte dell'input: il tuo obiettivo è proprio quello di trovarle.

Una volta trovate le N *golden nice lines*, dovrai chiamare la funzione:

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

Dove $a[i]$ e $b[i]$ dovranno descrivere la i -esima *golden nice line*, con $0 \leq i < N$. È permesso restituire le rette in qualsiasi ordine.

Assunzioni

- $1 \leq N \leq 100$
- $-10\,000 \leq a_i, b_i \leq 10\,000$
- Le rette non sono parallele.

¹La distanza euclidea tra un punto e una retta è la lunghezza del più corto segmento che connette il punto e la retta.

Assegnazione del punteggio

Il punteggio per un certo caso di test è calcolato nel modo seguente:

- Sia Q il numero di volte che la funzione `query` è stata chiamata.
- Se $Q > Q_{max}$, o se le *golden nice lines* non sono state riportate correttamente, il punteggio è 0.
- Se $Q \leq Q_{min}$, il punteggio è 1.
- Altrimenti, il punteggio è: $1 - 0.7 \cdot \frac{Q - Q_{min}}{Q_{max} - Q_{min}}$.

Il punteggio relativo a un subtask è calcolato prendendo il punteggio **minimo** ottenuto da un caso di test appartenente al subtask, e moltiplicando tale punteggio per il numero totale di punti del subtask.

Subtask 1 (11 punti)

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

Subtask 2 (13 punti)

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

Subtask 3 (7 punti)

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

Subtask 4 (19 punti)

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

Subtask 5 (23 punti)

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

Subtask 6 (27 punti)

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

Esempi

Chiamate fatte dal grader	Chiamate fatte dalla soluzione
<pre>solve(/* subtask_id = */ 1, /* N = */ 1)</pre>	<pre>query(0, 0) restituisce 0 query(1, 1) restituisce 0 the_lines_are(/* a = */ {1}, /* b = */ {0})</pre>