

Problem Nice Lines

C header: nice_lines_c.h
C++ header: nice_lines.h

Піратська принцеса Роксета прибула до таємного отсрова у Ремейському архіпелазі. Кажуть, що тут захований відомий скарб — *золоті чудові прямі*.

Таємний острів це квадрат, 2×10^{12} на 2×10^{12} метрів у довжину і висоту. Кожну точку на острові можна описати за допомогою декартових координат, де $(0,0)$ буде у центрі, і обидві осі симетрії паралельні сторонам острова.

На острові заховані N *золотих чудових прямих*. i -та пряма для $0 \leq i < N$ займає множину дійсних чисел (x, y) описаних рівнянням прямої $y = a_i x + b_i$.

Роксета може використати особливий пристрій — *прямометр*. Для будь-якої точки p на острові, *прямометр* порахує суму відстаней¹ від точки p до кожної з N *золотих чудових прямих*.

На жаль, *прямометр* має обмежену кількість використань. Чи ви допоможете Роксеті знайти скарб з достатньо малою кількістю використань *прямометра*?

Протокол взаємодії

Учасник має імплементувати одну функцію:

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

Ця функція буде викликана **рівно один раз** на початку взаємодії. N — це кількість *золотих чудових прямих* захованих на острові.

Ця функція може викликати іншу функцію, але **не більше ніж Q_{max} разів**:

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

Учасник має викликати цю функцію тільки з такими аргументами, що $-10^{12} \leq x, y \leq 10^{12}$.

Вона повертає відповідь *прямометра*, застосованого до точки із декартовими координатами (x, y) — тобто сума відстаней від точки (x, y) до кожної з N *золотих чудових прямих*. Зауважте, що самі *золоті чудові прямі* не будуть відомі, оскільки завдання учасника це їх пошук.

Коли учасник знайде N *золотих чудових прямих*, він має викликати функцію:

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

Де $a[i]$ та $b[i]$ мають описувати i -ту *золоту чудову пряму*, для $0 \leq i < N$. Учасники можуть повернути прямі у будь-якому порядку.

Обмеження

- $1 \leq N \leq 100$
- $-10\ 000 \leq a_i, b_i \leq 10\ 000$

¹Евклідова відстань між точкою і прямою це довжина найкоротшого відрізка який дотикається і точки, і прямої.

- Прямі не паралельні.

Оцінювання

Щоб порахувати оцінку для теста, виконуються наступні кроки:

- Нехай Q це кількість викликів функції `query`.
- Якщо $Q > Q_{max}$, або *золоті чудові прямі* були неправильно зазначені, тоді оцінка для тесту дорівнюватиме 0.
- Якщо $Q \leq Q_{min}$, тоді оцінка для тесту дорівнюватиме 1.
- Інакше, оцінка для тесту дорівнюватиме $1 - 0.7 \cdot \frac{Q - Q_{min}}{Q_{max} - Q_{min}}$.

Щоб порахувати оцінку для підзадачі, береться мінімальна оцінка з кожного з тестів у підзадачі і множиться на повну кількість балів у підзадачі.

Підзадача 1 (11 балів)

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

Підзадача 2 (13 балів)

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

Підзадача 3 (7 балів)

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

Підзадача 4 (19 балів)

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

Підзадача 5 (23 балів)

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

Підзадача 6 (27 балів)

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

Приклад

Виклики журі	Виклики учасника
<pre>solve(/* subtask_id = */ 1, /* N = */ 1)</pre>	<pre>query(0, 0) повертає 0 query(1, 1) повертає 0 the_lines_are(/* a = */ {1}, /* b = */ {0})</pre>