Problem Nice Lines

      C header: `nice_lines_c.h`

      C++ header: `nice_lines.h`

Roxette the pirate princess has arrived to the secret island in the Remeian archipelago. There, a famous treasure, the *golden nice lines* is rumoured to be buried.

The secret island is a square, 2×10^{12} by 2×10^{12} meters long and tall. Any point on the island is described using Cartesian coordinates, with $(0, 0)$ being at the center, and the two axes being parallel to its sides.

There are N *golden nice lines* buried on the island. The ith one for $0 \leq i < N$ occupies the set of all real-valued points (x, y) described by the linear equation $y = a_i x + b_i$.

Roxette can use a special device, called a *lineometer*. Given any point p on the island, the *lineometer* will compute the sum of the distances[1] from point p to each of the N *golden nice lines*.

Unfortunately, the *lineometer* has a limited number of uses. Can you help Roxette find the treasure with a small enough number of *lineometer* uses?

Interaction protocol

The contestant must implement one function:

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

This function will be called **exactly once**, at the beginning of the interaction. N is the number of *golden nice lines* hidden on the island.

This function is able to call another function, but **no more than Q_{max} times**:

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

The contestant must only call this function with arguments such that $-10^{12} \leq x, y \leq 10^{12}$.

It returns the result of the *lineometer* when applied to a point with Cartesian coordinates (x, y) – i.e. the sum of the distances from point (x, y) to each of the N *golden nice lines*. Note that the *golden nice lines* themselves will not be provided, as it is the contestant's objective to find them.

When the contestant finds the N *golden nice lines*, they must call the function:

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

Where $a[i]$ and $b[i]$ must describe the ith *golden nice line*, for $0 \leq i < N$. The contestant may return the lines in any order.

Constraints

- $1 \leq N \leq 100$

- $-10\ 000 \leq a_i, b_i \leq 10\ 000$

---

[1]The Euclidean distance between a point and a line is the length of the shortest line segment that touches both the line and the point.

- No two lines are parallel.

Scoring

To compute the score for a test, proceed as follows:

- Let Q be the number of times the `query` function has been called.

- If $Q > Q_{max}$, or if the *golden nice lines* have not been correctly reported, then the score for the test will be 0.

- If $Q \leq Q_{min}$, then the score for the test will be 1.

- Otherwise, the score for the test will be $1 - 0.7 \cdot \frac{Q - Q_{min}}{Q_{max} - Q_{min}}$.

To compute the score for a subtask, take the minimum score awarded for each of the tests in that subtask and then multiply it by the total number of points for the subtask.

Subtask 1 (11 points)

- $N = 1$

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

Subtask 2 (13 points)

- $N = 2$

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

Subtask 3 (7 points)

- $N = 3$

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

Subtask 4 (19 points)

- $-500 \leq a_i, b_i \leq 500$

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

Subtask 5 (23 points)

- $N \leq 30$

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

Subtask 6 (27 points)

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

Example

Committee calls	Contestant calls
solve(/* subtask_id = */ 1, /* N = */ 1)	query(0, 0) returns 0 query(1, 1) returns 0 the_lines_are(/* a = */ $\{1\}$, /* b = */ $\{0\}$)