

Board Meeting

Problem

Note that it is not necessary to know anything about the rules of chess to solve this problem.

There are N kings on an infinite chessboard (two-dimensional grid), located in cells with coordinates (X_1, Y_1) , (X_2, Y_2) , ..., (X_N, Y_N) . Both N and the kings' coordinates are unknown to you. However, you do know the following things:

- N is at least 1 and at most N_{\max} .
- No king's coordinates (X or Y) have an absolute value exceeding M .
- The N kings are located in N different cells.

The kings want to meet in a single cell of the board. If some cell (X, Y) were to be chosen as the meeting cell, then in order to get there, the i -th king would use a number of moves equal to the maximum of the absolute values of the differences of coordinates between its cell and the meeting cell: $\max(|X - X_i|, |Y - Y_i|)$. The total number of moves used by all kings is thus equal to the sum of those maximums over all values of i . Note that it is not relevant to this problem exactly how the kings move on the board — only the source and destination cells matter, and the number of moves can always be computed using the above formula.

This problem has two phases. In the first phase, you may repeatedly do the following: propose a meeting location (A, B) (with each of A and B between $-10 \times M$ and $10 \times M$, inclusive), and have the judge tell you the total number of moves the kings would use to get there — the sum (over all i) of $\max(|X_i - A|, |Y_i - B|)$. You can have at most R such exchanges with the judge, choosing your values of A and B each time. Note that the kings do not actually move, so their locations (X_i, Y_i) stay the same for all requests within one test case.

In the second phase, the roles are swapped: the judge gives you a meeting cell location (C, D) (with each of C and D between $-10 \times M$ and $10 \times M$, inclusive), and you must respond with the total number of moves the kings would use to get there, assuming that the kings are in the same locations as in the first phase. There are at most R such exchanges, and you must correctly respond to all of the judge's requests.

Input and output

This is an interactive problem. You should make sure you have read the information in the Interactive Problems section of our [FAQ](#).

Initially, your program should read a single line containing four integers T , N_{\max} , M and R : the number of test cases, the maximum number of kings, the maximum absolute value for any coordinate for any king, and the maximum number of requests per phase, respectively. (Note that the values of M and R are fixed, and are provided as input only for convenience; see the Limits section for more details.) Then, you need to process T test cases.

In each test case, there are two phases. In the first phase, the i -th exchange is as follows:

- Your program sends one line containing two integers A_i and B_i , representing the x and y coordinates of a cell.
 - Both A_i and B_i must be between $-10 \times M$ and $10 \times M$, inclusive.
- The judge responds with one line containing a single integer: the total number of moves the kings need to use to get from their unknown locations to your cell.

You may initiate at most R such exchanges in this phase. If you make more than R exchanges, or send a request that the judge can not parse or is out of bounds, the judge responds with one line with a single string `ERROR`.

To end the first phase and switch to the second phase, you must send one line with the string `READY` (the case does not matter), to which the judge responds with the first request of the second phase.

In the second phase, the i -th exchange is as follows:

- The judge sends one line containing two integers C_i and D_i , representing the x and y coordinates of a cell.
 - Each of C_i and D_i will be between $-10 \times M$ and $10 \times M$, inclusive.
- Your program must respond with one line containing a single integer: the total number of moves the kings would need to use to get to the given cell.

The judge is guaranteed to send at least 1 and at most R such requests. If you send an answer that is incorrect or unparseable, the judge responds with `ERROR` as described above. If you answer all of the requests correctly, the judge sends one line with a single string `DONE`, at which point your program should initiate the next test case, or terminate with no error if all T test cases have been handled.

After the judge sends a line with `ERROR`, it does not send any other output. If your program continues to wait for the judge after receiving `ERROR`, your program will time out, resulting in a Time Limit Exceeded error. Notice that it is your responsibility to have your program exit in time to receive a Wrong Answer judgment instead of a Time Limit Exceeded error. As usual, if the memory limit is exceeded, or your program gets a runtime error, you will receive the appropriate judgment.

The number and location of the kings, as well as the number and positions of the requests that the judge sends during the second phases, are chosen before any exchanges occur.

Limits

Time limit: 60 seconds per test set.

Note that a program that just makes valid exchanges with the judge (and does no other processing) takes the following time in our environment: ~13 seconds for C++, ~24 seconds for Java, ~19 seconds for Python and Go.

Memory limit: 1GB.

$1 \leq T \leq 15$.

$M = 10^6$.

$-M \leq X_i \leq M$, for all i .

$-M \leq Y_i \leq M$, for all i .

The pairs (X_i, Y_i) are distinct.

$-10 \times M \leq C_i \leq 10 \times M$, for all i .

$-10 \times M \leq D_i \leq 10 \times M$, for all i .

$R = 1000$.

Test set 1 (Visible)

$N_{\max} = 1$.

Test set 2 (Hidden)

$N_{\max} = 10$.

Testing Tool

You can use this testing tool to test locally or on our platform. To test locally, you will need to run the tool in parallel with your code; you can use our [interactive runner](#) for that. For more information, read the instructions in comments in that file, and also check out the [Interactive Problems section](#) of the FAQ.

Instructions for the testing tool are included in comments within the tool. We encourage you to add your own test cases. Please be advised that although the testing tool is intended to simulate the judging system, it is **NOT** the real judging system and might behave differently. If your code passes the testing tool but fails the real judge, please check the [Coding section](#) of the FAQ to make sure that you are using the same compiler as us.

[Download testing tool](#)

Sample Interaction

Note that the following sample interaction is for test set 1, in which there is always exactly one king.

```
// Suppose that the judge has decided that in the first test case, the king
// is at the coordinates (1, -2), and the requests will be (5, -1) and
// (7, 7).
t, nmax, m, r = readline_int_list()    // Reads 10 1 1000000 1000
// Our solution decides (for whatever reason) to check (3, 3) first.
println 3 3 to stdout
flush stdout
result = readline_int()                // Reads 5
// Our solution now decides (for whatever reason) to check (2, 0).
println 2 0 to stdout
flush stdout
result = readline_int()                // Reads 2
// Our solution concludes that the king is at (3, -2), which is consistent
// with the observed information so far, but unfortunately not correct.
// Our solution moves on to the request phase.
println READY to stdout
request_line = readline()              // Reads 5 -1
println 2 to stdout                    // Wrong answer!
request_line = readline()              // Reads ERROR
exit                                  // exits to avoid an ambiguous TLE error
```