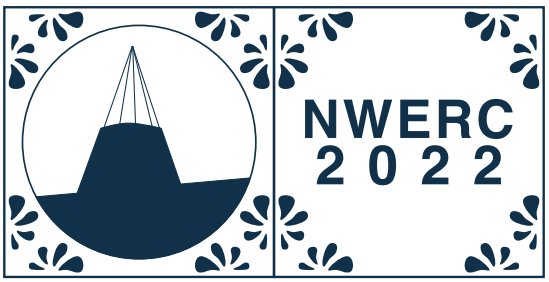
Test Session for the Northwestern Europe Regional Contest 2022

NWERC 2022 Test Session

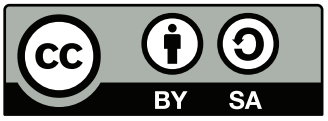
November 26, 2022



TU Delft Delft University of Technology

Test Session Problems

A An Interactive Problem
B Brothers in Arms
C Cup Covering



This page is intentionally left blank.

Problem A
An Interactive Problem
Time limit: 2 seconds

Do not worry, they made up with hot cocoa and stroopwafels afterwards. Generated using imgflip.com

Troubles in the jury! The contest day of NWERC is tomorrow already, and they still need a nice interactive problem for the problem set. During this test session, the jury have gathered in an emergency meeting and started discussing problem ideas. All of the n^2 jury members have an idea for an interactive problem, and even after hours of arguing, they cannot decide which idea is the nicest. To resolve the discussion as quickly as possible, they decide to draw lots.

Each member of the jury draws a lot from a collection of unique numbers. Because the number of jury members is square and they like regular shapes, they place all the lots face-down on a table, laid out in the shape of a square. They task you to be the independent lot-drawer, to see which jury member has drawn the highest number. To make sure you are truly independent, you are not allowed to see the table, hence you to pick the lots on the table by shouting coordinates. They are in a hurry, though, so you need to find the lot with the highest number in at most $n^2 + 100$ commands. You do not know the size of the square beforehand, but you do know that n is at least 1 and at most 100.

Interaction

This is an interactive problem. Your submission will be run against an *interactor*, which reads from the standard output of your submission and writes to the standard input of your submission. This interaction needs to follow a specific protocol:

Your program should make at most $n^2 + 100$ queries to find the lot with the highest value. Each query is made by printing one line of the form "`?` x y" ($1 \leq x, y \leq 1000$), the coordinate of the lot you want to uncover. Note that the top-left corner of the square of lots is always at $(1, 1)$. The interactor will respond with either an integer v ($1 \leq v \leq 10^9$), indicating the value on the drawn lot, or the string "`ArrayIndexOutOfBoundsException`" if the chosen coordinate is outside the square.

When you have determined the lot with the highest value v, print one line of the form "`!` v", after which the interaction will stop. Printing the answer does not count as a query.

If there are multiple valid solutions, you may output any one of them.

Make sure you flush the buffer after each write.

A testing tool is provided to help you develop your solution.

The interactor is not adaptive; the size of the square is determined by the interactor before any interaction takes place. Using more than $n^2 + 100$ queries will result in a wrong answer verdict.

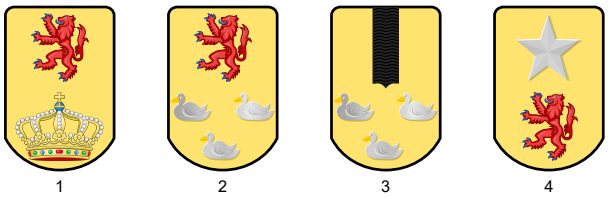
Sample Interaction 1

Read	Write
	`? 1 1`
`314`	
	`? 4 5`
`ArrayIndexOutOfBoundsException`	
	`? 2 2`
`123456`	
	`? 1 3`
`26`	
	`? 2 3`
`11`	
	`? 3 3`
`2022`	
	`? 1 2`
`1337`	
	`? 1 4`
`ArrayIndexOutOfBoundsException`	
	`? 3 2`
`1`	
	`? 3 1`
`42`	
	`? 2 1`
`999`	
	`! 123456`

Problem B
Brothers in Arms
Time limit: 8 seconds

In medieval times, keeping track of the relationships between cities was extremely difficult, since most cities did not have access to the internet[citation needed]. However, it was possible to determine whether two cities were friendly with each other by examining their coats of arms. In those days, every coat of arms showed two symbols: one at the top, and one at the bottom. If two cities have an equal symbol at the top or they have an equal symbol at the bottom, they are friendly.

Following the saying "the friends of my friends are my friends", two cities c_0 and c_f can be indirectly friendly if there exist cities $c_1, \ldots, c_{f-1}$ such that c_k is friendly with c_{k+1} for $0 \leq k < f$. If c_0 and c_f are different and indirectly friendly, then we say that the friendship degree of these cities is the smallest possible f following this definition. See Figure B.1 for an example.



Parts of these coats of arms are CC BY-SA 4.0 on Wikimedia Commons.

Figure B.1: Illustration of Sample Input 1. Cities 1 and 2 are directly friendly, as well as cities 2 and 3. Cities 1 and 3 have a friendship degree of 2, because they are indirectly friendly via city 2. City 4 is not (indirectly) friendly with any other city.

You are given a list of coats of arms and a list of queries. For every query, determine the friendship degree of the two given cities.

Input

The input consists of:

- One line with two integers n and s ($2 \leq n \leq 90\,000$, $2 \leq s \leq 300$), the number of cities and the number of symbols that may appear on the coat of arms of some city.
- n lines, the ith of which consists of two integers t_i and b_i ($1 \leq t_i, b_i \leq s$). t_i is the symbol on the top side of the coat of arms of the ith city, and b_i is the symbol on the bottom side of the coat of arms of the ith city. If $i \neq j$, then $t_i \neq t_j$ or $b_i \neq b_j$.
- One line with an integer q ($1 \leq q \leq 10^5$), the number of queries.
- q lines, the ith of which contains two integers c and d ($1 \leq c, d \leq n$, $c \neq d$), two cities for which you should calculate the friendship degree.

Output

For every query, output an integer stating the friendship degree of the two cities, or -1 if the two cities are not (indirectly) friendly.

Sample Input 1

```
4 5
1 2
3 2
3 4
5 1
4
1 2
2 3
1 3
1 4
```

Sample Output 1

```
1
1
2
-1
```

Sample Input 2

```
7 5
1 1
2 1
2 2
3 2
4 2
3 5
3 4
6
1 2
1 3
1 7
1 4
1 5
6 7
```

Sample Output 2

```
1
2
4
3
3
1
```

Sample Input 3

```
2 4
1 2
3 4
1
1 2
```

Sample Output 3

```
-1
```

Problem C
Cup Covering
Time limit: 1 second

A stroopwafel on a slightly too small cup.

Janneke is addicted to Dutch stroopwafels. She could eat them the whole day, every day. What she loves most during the colder seasons, is to put a stroopwafel on top of a cup full of hot, steaming cocoa. This way, the stroopwafel warms and softens and the caramel melts, leaving a gooey but delicious mess between her fingers when she picks it up.

For Janneke, this is pure heaven. Or well, it could be if she could just find the perfect cup for the round stroopwafels. Instead, she is left with unsatisfying stroopwafel experiences time and time again. Either the opening of the cup is too small for her stroopwafel and the edges stay hard and cold or – even worse – the opening is too large and the stroopwafel just falls in with a splash, becoming soggy and disgusting.

Janneke finally has had enough and decides to take matters into her own hands. She establishes the *Ideal Cup Production Company* (ICPC) which produces the ideal stroopwafel cups. Customers can simply tell her the area of the round stroopwafel they prefer and she delivers a cup with an opening that is perfectly covered by the stroopwafel. For the production, she needs to determine the diameter of the cup opening first. Soon, she will have orders from all over the world which she can barely keep up with, so she might need a little help here.

Input

The input consists of:

- One line with an integer a ($0 < a \leq 10^{15}$), the area of the round stroopwafel in cm^2.

Output

Output the diameter of the ideal cup in centimetres such that a stroopwafel with area a covers it perfectly. Note that the rim of the cup is so thin that it can be neglected.

Your answer should have an absolute or relative error of at most 10^{-9}.

Sample Input 1	Sample Output 1
42	7.3127327914

Sample Input 2	Sample Output 2
20221126	5074.0848694393