German Collegiate Programming Contest Practice Session 2022

June 15th

Problems

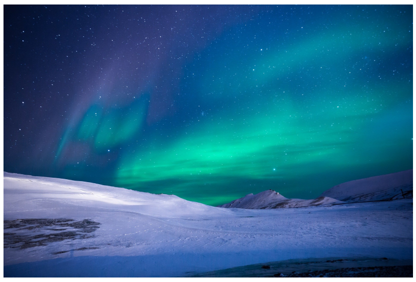
A Aurora Borealis

B Basalt Breakdown

C Counterfeit Coin

This page is intentionally left (almost) blank.

Problem A: Aurora Borealis


Aurora Borealis, CC0 Public Domain

Tonight is finally your chance to experience one of the most mystifying phenomena the earth has to offer: the Aurora Borealis. A week ago you begged your parents to let you go out to the countryside this evening, and luckily they gave you the green light. You put on warm clothes, fill your flask with hot coffee, and prepare a quick snack. It seems like conditions are perfect; there is only a single cloud on the horizon and you look forward to the beautiful prismatic display.

However, reality once agains rears its ugly head. Every time you try to look at the Aurora Borealis, this stupid single cloud is in the way! You cannot believe the bad luck you are having, but you do not get a single look at the Aurora Borealis that night.

Disgruntled and disillusioned you head back home and try to make the best of the short amount of sleep you have left. Unfortunately, anger-induced insomnia gets the best of you and you start thinking about this annoying cloud. You feel like you got disproportionally unlucky, and in order to get some peace of mind you decide to calculate how large the cloud had to be in order to foil your sightseeing plans so effectively.

You model the sky as the real line, and the cloud as a closed interval, obscuring from view any point in this interval. You assume the speed of the cloud can never exceed 1 m/s. Whenever you looked at the sky, you wrote down the exact time it was and the exact position you looked at. Given this information, what is the smallest length the cloud could have so that the Aurora Borealis remained hidden to you?

Input

The input consists of:

- A line containing the integer n ($1 \leq n \leq 3 \cdot 10^5$), the number of times you looked at the sky.
- n lines, each with two integers t and x ($0 \leq t, x \leq 10^{12}$), the time and position of one of the attempts to see the Aurora.

The times in the input are all unique.

Output

Output the minimum possible length of the cloud that would be able to keep you from seeing the Aurora. It can be proven that the result is always an integer.

Sample Input 1

```
5
3 1
5 5
2 3
8 9
0 0
```

Sample Output 1

```
3
```

Sample Input 2

```
3
1 1
5 2
10 1
```

Sample Output 2

```
0
```

Sample Input 3

```
5
0 0
1000000000 1000000000
110000000000 80000000000
10000000000 10000000000
100000000000 100000000000
```

Sample Output 3

```
10000000000
```

Problem B: Basalt Breakdown

Svartifoss by Piotr Wojtkowski, CC0 Public Domain

One of Iceland's most popular attractions is *Svartifoss* ("black waterfall"). Its name derives from the black hexagonal basalt columns that frame the waterfall on either side. Originally formed from cooling lava, centuries of erosion have shaped the columns into their characteristic shape.

A group of geologists at RU went on an excursion to Svartifoss. They took some probes and performed various measurements on the hexagonal rocks that have broken off the basalt walls.

Just as they return to RU, they realise that they have forgotten a crucial measurement. They have determined the area of the hexagonal face, but they did not write down what its perimeter was. Assuming that the face has the shape of a perfect regular hexagon, help the geologists compute the perimeter.

Input

The input consists of:

- One line with an integer a ($1 \leq a \leq 10^{18}$), the area of the hexagonal rock face in square centimetres.

Output

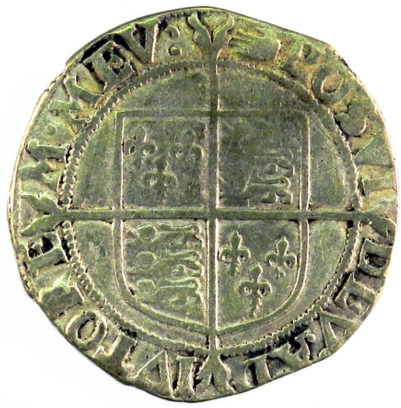
Output the perimeter of the rock face in centimetres. Your answer should have an absolute or relative error of at most 10^{-6}.

Sample Input 1	Sample Output 1
`50`	`26.32148026`

Sample Input 2	Sample Output 2
`1234`	`130.76240122`

This page is intentionally left (almost) blank.

Problem C: Counterfeit Coin



Counterfeit Shilling of Elizabeth I, CC BY-SA 3.0

You are a market salesman. Every week you sell your wares in a beautiful authentic market stall. People travel from faraway lands to buy your signature product: fermented shark. Of course, such a delicious product should only be exchanged for a vast amount of shining, valuable coins.

Sometimes customers try to fool you by paying with counterfeit coins. You always spot them, of course; you take out your trusty scale and weigh the coins to determine if any of them are of different weight.

In this particular instance, a rather notorious customer by the name of Peter Rog Rammer, famous for his "off-by-one errors", has just paid for a fermented shark dish using n seemingly identical coins. However, you are sure that there is one coin that is not quite the same as the others. As the line of customers is getting rather lengthy, you do not want to spend too much time finding the odd one out!

Interaction

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

- The interactor first sends an integer n ($3 \leq n \leq 2 \cdot 10^5$).
- Your submission then repeatedly sends two integers x and y ($1 \leq x, y \leq n$, $x \neq y$) preceded by "`?`", indicating that you want to compare the xth and yth coin.
- The interactor replies with a string:
  - "`lighter`" if the xth coin weighs less than the yth coin.
  - "`heavier`" if the xth coin weighs more than the yth coin.
  - "`equal`" if the xth coin weighs the same as the yth coin.
- When you are ready to print the answer, print "`!`" followed by a single integer x ($1 \leq x \leq n$), indicating that the xth coin is the coin that does not weigh the same as the others.

You may use at most $\lceil n/2 \rceil$ moves. Printing the answer does not count as a move.

Note that for testing purposes, the index of the counterfeit coin is not necessarily fixed at the start of the interaction, but may be set by the interactor at any point (but always in a way that is consistent with its answers to previous queries).

Make sure you flush the buffer after each write.

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

Read	Sample Interaction 1	Write
`5`		
		`? 1 5`
`equal`		
		`? 4 3`
`lighter`		
		`? 2 4`
`heavier`		
		`! 4`