Problem A. Hidden Integer

Time Limit 1000 ms

Mem Limit 524288 kB

There is a hidden integer x. Your task is to find the value of x.

To do this, you can ask questions: you can choose an integer y and you will be told if $y < x$.

Interaction

This is an interactive problem. Your code will interact with the grader using standard input and output. You can start asking questions right away.

On your turn, you can print one of the following:

- "? y", where $1 \leq y \leq 10^9$: ask if $y < x$. The grader will return `YES` if $y < x$ and `NO` otherwise.
- "! x": report that the hidden integer is x. Your program must terminate after this.

Each line should be followed by a line break. You must make sure the output gets flushed after printing each line.

Constraints

- $1 \leq x \leq 10^9$
- you can ask at most 30 questions of type ?

Example

```
? 3
YES
? 6
YES
? 7
NO
! 7
```

Problem B. Hidden Permutation

Time Limit	1000 ms
Mem Limit	524288 kB

There is a hidden permutation $a_1, a_2, \ldots, a_n$ of integers $1, 2, \ldots, n$. Your task is to find this permutation.

To do this, you can ask questions: you can choose two indices i and j and you will be told if $a_i < a_j$.

Interaction

This is an interactive problem. Your code will interact with the grader using standard input and output. You should start by reading a single integer n: the length of the permutation.

On your turn, you can print one of the following:

- "? i j", where $1 \leq i, j \leq n$: ask if $a_i < a_j$. The grader will return `YES` if $a_i < a_j$ and `NO` otherwise.
- "! a_1 $a_2 \ldots a_n$": report that the hidden permutation is $a_1, a_2, \ldots, a_n$. Your program must terminate after this.

Each line should be followed by a line break. You must make sure the output gets flushed after printing each line.

Constraints

- $1 \leq n \leq 1000$
- you can ask at most 10^4 questions of type ?

Example

```
3
? 3 2
NO
? 3 1
YES
! 3 1 2
```

Explanation: The hidden permutation is $[3, 1, 2]$. The first question asks if $a_3 < a_2$ which is false, so the answer is `NO`. The second question asks if $a_3 < a_1$ which is true, so the answer is `YES`.

Problem C. K-th Highest Score

Time Limit	1000 ms
Mem Limit	524288 kB

There were n coders from Finland and n coders from Sweden in a programming contest. It turned out that after the contest, each coder had a distinct score.

Your task is to find the k-th highest score in the contest.

To do this, you can ask questions: you can choose a country (Finland or Sweden) and an integer i and you will be told the i-th highest score for the chosen country.

Interaction

This is an interactive problem. Your code will interact with the grader using standard input and output. You should start by reading two integers n and k.

On your turn, you can print one of the following:

- "F i", where $1 \le i \le n$: ask the i-th highest score for Finland.
- "S i", where $1 \le i \le n$: ask the i-th highest score for Sweden.
- "! s": report that the k-th highest score is s. Your program must terminate after this.

Each line should be followed by a line break. You must make sure the output gets flushed after printing each line.

Constraints

- $1 \le n \le 10^5$
- $1 \le k \le 2n$
- each score is between 1 and 10^9
- you can ask at most 100 queries of the first two types in total

Example

```
3 1
F 1
9
S 1
8
! 9
```

Explanation: The scores for Finland are $[9, 4, 3]$ and the scores for Sweden are $[8, 6, 1]$. Since $k = 1$, the task is to find the highest score overall, which in this case is 9.

Problem D. Permuted Binary Strings

Time Limit 1000 ms

Mem Limit 524288 kB

There is a hidden permutation $a_1, a_2, \ldots, a_n$ of integers $1, 2, \ldots, n$. Your task is to find this permutation.

To do this, you can ask questions: you can choose a binary string $b_1 b_2 \ldots b_n$ and you will receive the binary string $b_{a_1} b_{a_2} \ldots b_{a_n}$.

Interaction

This is an interactive problem. Your code will interact with the grader using standard input and output. You should start by reading a single integer n: the length of the permutation.

On your turn, you can print one of the following:

- "? $b_1 b_2 \ldots b_n$", where $b_i \in \{0, 1\}$: The grader will return the binary string $b_{a_1} b_{a_2} \ldots b_{a_n}$.
- "! $a_1\ a_2 \ldots a_n$": report that the hidden permutation is $a_1, a_2, \ldots, a_n$. Your program must terminate after this.

Each line should be followed by a line break. You must make sure the output gets flushed after printing each line.

Constraints

- $1 \le n \le 1000$
- you can ask at most 10 questions of type ?

Example

```
3
? 100
100
? 010
001
? 001
010
! 1 3 2
```

Explanation: The hidden permutation is $[1, 3, 2]$. In the first question $b_1b_2b_3 = 100$ and the grader returns $b_{a_1}b_{a_2}b_{a_3} = b_1b_3b_2 = 100$. In the second question $b_1b_2b_3 = 010$ and the grader returns $b_1b_3b_2 = 001$.

Problem E. Colored Chairs

Time Limit 1000 ms

Mem Limit 524288 kB

There are n chairs arranged in a circle. Each chair is either red or blue. The chairs are numbered $1, 2, \ldots, n$; chairs i and $i+1$ are next to each other for all $1 \leq i \leq n$. Here chair $n+1$ refers to chair 1.

Your task is to find two chairs that have the same color and are next to each other.

To do this, you can ask questions: you can choose a chair and you will be told the color of that chair.

Interaction

This is an interactive problem. Your code will interact with the grader using standard input and output. You should start by reading a single integer n: the number of chairs.

On your turn, you can print one of the following:

- "? i", where $1 \leq i \leq n$: ask the color of chair i. The grader will return `R` or `B` for red or blue.
- "! i": report that chairs i and $i+1$ have the same color. Your program must terminate after this.

Each line should be followed by a line break. You must make sure the output gets flushed after printing each line.

Constraints

- $3 \leq n \leq 2 \cdot 10^5$, n is odd
- you can ask at most 20 questions of type ?

Example

```
5
? 1
R
? 2
B
? 3
```

```
B
! 2
```

Problem F. Inversion Sorting

Time Limit 1000 ms
Mem Limit 524288 kB

There is a hidden permutation $a_1, a_2, \ldots, a_n$ of integers $1, 2, \ldots, n$. Your task is to sort the permutation by reversing subarrays.

On each turn, you can reverse a subarray of the permutation. After that, you will be reported the number of inversions in the permutation. If the number of inversions is 0 (i.e., the permutation is sorted), you win.

Interaction

This is an interactive problem. Your code will interact with the grader using standard input and output. You should start by reading a single integer n: the length of the permutation.

On your turn, print two integers i and j: reverse the subarray between indices i and j.

After this, the next input line has a single integer: the number of inversions after the operation. If the number is 0, you win and your program must terminate after this.

Constraints

- $1 \leq n \leq 1000$
- you can make at most $4n$ operations

Example

```
3
1 2
1
2 3
0
```

Explanation: Here the initial permutation is $[3, 1, 2]$. After the first operation the permutation is $[1, 3, 2]$ and the number of inversions is 1. After the second operation the permutation is $[1, 2, 3]$ and the number of inversions is 0.