Team Test 1 Problems

A. Are You Free Now?

1 second, 256 megabytes

Your friend Taylor is recently in between opportunities and she decided to take this time to go live in Hong Kong for a few months before she comes back to the states to hunt for a job. Unfortunately, Hong Kong is so far away from RPI that time zone differences exist. You want to know what time it is for Taylor so you could decide if it's a nice time to call her to talk about your latest crush. Write a program to convert time at RPI to time at Hong Kong.

Note: Hong Kong is currently 12 hours ahead of RPI.

Input
A string consisting of weekday followed by military time at RPI

Output
A string consisting of weekday followed by military time at Hong Kong

input
```
Saturday 0034
```
output
```
Saturday 1234
```

Weekdays are: "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", and "Saturday"

B. But some numbers are missing

1 second, 256 megabytes

This is an interactive problem. You have to use a `flush` *operation right after printing each line. For example, in C++ you should use the function* `fflush(stdout)` *and in Python —* `sys.stdout.flush()`.

Your friend Chloe has a set $S = \{1, 2, \dots, N\}$ and she decided to randomly remove k numbers from it. Since she's like being mysterious and what not, she is not going to directly tell you which k numbers are removed. Instead, she makes it a guessing game for you to figure out the missing numbers. In each question, you will be able to ask the x-th smallest element of S. To prevent you from using bruteforce to figure out all k numbers, Chloe decided that you could ask at most **800** times.

Input
Use standard input to read the responses to the queries.

The first line contains two integers N and k ($1 \le n, k \le 10^5$, additionally $k \le N$)

Following lines will contain responses to your queries, an integer representing the x-th smallest element in S. If there doesn't exist x elements in S, -1 will be given instead. The $i + 1$-th line is a response to your i-th query.

Output
To make the queries your program must use standard output.

If your program wants to make a query, print a single question mark followed by integer x_i, one query per line (do not forget "*end of line*" after each x_i). After printing each line your program must perform operation `flush`. The testing system will allow you to read the response on the query only after your program print the query for the system and perform `flush` operation.

If your program wants to make a guess of what the missing numbers are, print a single question mark followed by k space separated integers (and an "*end of line*") and **terminate** your program.

input
```
5 3

2

4
```
output
```

? 1

? 2

! 1 3 5
```

C. Clap 7

1 second, 256 megabytes

Clap 7 is a game where players will have to clap instead of say out the number if the number is divisible by 7 or contains the digit 7.

Your friend Chloe is babysitting her cousins and her cousins decided to make her the referee of the game. Since she's too busy watching TikTok edits of Taylor Swift, help Chloe to find the next 10 numbers where the player has to clap.

Input
A single integer N ($1 \le N \le 10^{18}$)

Output
10 lines consisting a single integer which is the next 10 numbers where a player should clap

input
```
7
```
output
```
14
17
21
27
28
35
37
42
47
49
```

D. Debug this?

1 second, 256 megabytes

Your teammate Chloe was trying to do the mock test to get familiar with questions written by the teaching staff. However, she has run into a bug she cannot fix, this results in her getting the WRONG ANSWER verdict. Since you also wanted to do well in the test, you decided to investigate into this issue.

You will be able to locate the problem statement for "A + B (Hard Version)" and Chloe's buggy code under contest material or at the end of the printed problem booklet.

Input

This problem is an output only task and would not contain any inputs.

Output

Output a valid input that will make Chloe's code produce a WRONG ANSWER verdict. You must follow the input format as described in the problem statement.

E. Easier game about permutation

1 second, 256 megabytes

The only difference between this question and "Game about permutation" is the constraint in N and k

Your friend Chloe recently had a lot of time on her hands and was trying to observe properties of permutations of $\{1, \dots, N\}$. She defined the **bumpiness** of an array A to be $\max(LIS(A), LDS(A))$ where $LIS(A)$ and $LDS(A)$ is the length of the longest increasing subsequence and the length of the longest decreasing subsequence of A.

She's particularly interested in what bumpiness is achievable for permutations of a fixed size N. Given an integer k, find a permutation of $\{1, \dots, N\}$ such that the bumpiness of said permutation is less than or equal to k.

Input

The first line contains 2 integers N, k. ($1 \le k \le N \le 10$)

Output

On the first line, output "`Possible`" (without quotes) if it's possible to find a permutation with bumpiness less than or equal to k. If it's impossible, then output "`Impossible`" (without quotes)

If it is possible to find a permutation with bumpiness less than or equal to k, on the second line output N space separated integers $p_1, \dots, p_N$ which represents the permutation matching the requirement

input
```
5 3
```
output
```
Possible
1 3 5 4 2
```

input
```
2 1
```
output
```
Impossible
```

F. Fibonacci Sum

2.5 seconds, 256 megabytes

Define the Fibonacci sequence as follows: $F_0 = 0, F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \ge 2$.

Write a program that calculates the following:

$$\sum_{i=L}^{R} F_i$$

Input

First line consists of an integer Q ($1 \le Q \le 10^5$). The next Q lines consists of two integers L_i and R_i ($1 \le L_i \le R_i \le 10^5$)

Output

The desired sum. Since the value could be very large, output the value mod $10^9 + 7$

input
```
2
0 2
3 5
```
output
```
2
10
```

G. Game about permutation

1 second, 256 megabytes

The only difference between this question and "Easier game about permutation" is the constraint in N and k

Your friend Chloe recently had a lot of time on her hands and was trying to observe properties of permutations of $\{1, \dots, N\}$. She defined the **bumpiness** of an array A to be $\max(LIS(A), LDS(A))$ where $LIS(A)$ and $LDS(A)$ is the length of the longest increasing subsequence and the length of the longest decreasing subsequence of A.

She's particularly interested in what bumpiness is achievable for permutations of a fixed size N. Given an integer k, find a permutation of $\{1, \dots, N\}$ such that the bumpiness of said permutation is less than or equal to k.

Input

The first line contains 2 integers N, k. ($1 \le k \le N \le 10^5$)

Output

On the first line, output "`Possible`" (without quotes) if it's possible to find a permutation with bumpiness less than or equal to k. If it's impossible, then output "`Impossible`" (without quotes)

If it is possible to find a permutation with bumpiness less than or equal to k, on the second line output N space separated integers $p_1, \dots, p_N$ which represents the permutation matching the requirement

input
```
5 3
```
output
```
Possible
1 3 5 4 2
```

input
```
2 1
```
output
```
Impossible
```

H. House Robber

1 second, 256 megabytes

You are a robber visiting the number line neighborhood. You know that there are a total of N houses in the neighborhood. The i-th house is at position x_i and has gems valued at v_i. You initially start at position 0 and moving left or right by 1 unit will cost 1 unit of energy. However, robbing a house will not cost any energy. You start with k units of energy, what is the maximum value of gems you could get?

Input

First line consists of two integers N and k ($1 \le N \le 10^5$, $1 \le k \le 10^9$). The next N lines consists of 2 integers, x_i and v_i ($-10^9 \le x_i \le 10^9, 0 \le v_i \le 10^9$)

Output

Output a single integer, the maximum value you could get.

input
```
2 3
-1 4
1 5
```
output
```
9
```

I. Is my permutation too bumpy?

1 second, 256 megabytes

Your friend Chloe recently had a lot of time on her hands and was trying to observe properties of permutations of $\{1, \dots, N\}$. She defined the **bumpiness** of an array A to be $\max(LIS(A), LDS(A))$ where $LIS(A)$ and $LDS(A)$ is the length of the longest increasing subsequence and the length of the longest decreasing subsequence of A.

Given a permutation $A = A_1, \dots, A_N$, find the permutation's bumpiness

Input

The first line contains an integer N ($1 \le N \le 1000$). The second line contains N space-separated integers $A_1, \dots, A_N$ ($1 \le A_i \le N$, where $A_i \ne A_j$ for all $i \ne j$), representing the permutation

Output

A single line containing the bumpiness of the permutation given.

input
```
3
1 2 3
```
output
```
3
```

J. Finding your partner

1 second, 256 megabytes

This is an interactive problem. You have to use a `flush` operation right after printing each line. For example, in C++ you should use the function `fflush(stdout)` and in Python — `sys.stdout.flush()`.

You and your partner were lining up for the newest attraction in your local amusement park, named Double Trouble. However, while waiting for your turn, you had the sudden urge to go to the bathroom. In order to get back to the queue, you must tell the cast member where your partner is exactly in the queue, or else you can't get back in.

In this problem, there is a queue of length n, where n is an odd number and is given to you at the beginning. Your partner is at position x which is between 1 and n.

Lucky for you, the queue has already been organized into pairs, i.e., the i-th person would be paired with either the $i-1$-th person or the $i+1$-th person for all i where $i \ne x$ and $1 \le i \le n$.

Since it would be ridiculous to be able to guess your partner's position without any information from your partner, the queue manager decides to let you ask a few queries about the queue. Each query is two integers a and b from 1 to n. Flush output stream after printing each query. There are two different responses the queue manager can provide:

- the string "`YES`" (without quotes), if a and b are of the same pair;
- the string "`NO`" (without quotes), if a and b are not of the same pair.

When you know the position of your partner, print string "`! x`", where x is the position of your partner, and **terminate your program normally** immediately after flushing the output stream.

Since the queue manager doesn't have much time to spare, your program is allowed to make no more than 16 queries (not including printing the answer).

Input

Use standard input to read the responses to the queries.

The first line contains an odd integer n ($1 \le n < 10^5$) — number of people in the queue.

Following lines will contain responses to your queries — strings "YES" or "NO". The i-th line is a response to your i-th query. When your program will guess the number print "`! x`", where x is the answer and terminate your program.

The testing system will allow you to read the response on the query only after your program print the query for the system and perform `flush` operation.

Output

To make the queries your program must use standard output.

Your program must print the queries — a single question mark followed by integer numbers a_i, b_i separated by space ($1 \le a_i, b_i \le n$), one query per line (do not forget "*end of line*" after each y_i). After printing each line your program must perform operation `flush`.

Each of the values a_i, b_i mean the query to the testing system. The response to the query will be given in the input file after you flush output. In case your program guessed the number x, print string "`! x`", where x — is the answer, and terminate your program.

input
```
5

YES

NO
```
output
```

? 1 2

? 3 4

! 3
```

K. A + B (Hard Version)

1 second, 256 megabytes

Somehow, someone was able to override the default `add` operation to the one described by the pseudocode shown below:

```
function add(a, b):
  start = time_now()
  sleep(a)
  sleep(b)
  return time_now() - start
```

Unfortunate for you, you don't have enough time to figure out how to revert that change and you just so happen to need to do a bunch of addition. Given n integers $a_1, a_2, \dots, a_n$, find the minimum time required to calculate the sum $\sum_{i=1}^{n} a_i$ using the overridden `add` function.

Input

The first line of input consists of 1 integer n $(1 \le n \le 10^5)$. The second line of input consists of n integers separated by spaces, where the i-th integer represents a_i $(1 \le a_i \le 10^9)$.

Output

Print the minimum time required to sum up all numbers.

input

```
3
1 2 4
```

output

```
10
```

input

```
5
1 1 1 1 1
```

output

```
12
```