A. Infinite Sequence

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Consider the infinite sequence of integers: $1, 1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5...$. The sequence is built in the following way: at first the number 1 is written out, then the numbers from 1 to 2, then the numbers from 1 to 3, then the numbers from 1 to 4 and so on. Note that the sequence contains numbers, not digits. For example number 10 first appears in the sequence in position 55 (the elements are numerated from one).

Find the number on the n-th position of the sequence.

Input

The only line contains integer n ($1 \le n \le 10^{14}$) — the position of the number to find.

Note that the given number is too large, so you should use 64-bit integer type to store it. In C++ you can use the `long long` integer type and in Java you can use `long` integer type.

Output

Print the element in the n-th position of the sequence (the elements are numerated from one).

Examples

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

input
```
5
```
output
```
2
```

input
```
10
```
output
```
4
```

input
```
55
```
output
```
10
```

input
```
56
```
output
```
1
```

B. The Time

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given the current time in 24-hour format `hh:mm`. Find and print the time after a minutes.

Note that you should find only the time after a minutes, see the examples to clarify the problem statement.

You can read more about 24-hour format here https://en.wikipedia.org/wiki/24-hour_clock.

Input

The first line contains the current time in the format `hh:mm` ($0 \le hh < 24, 0 \le mm < 60$). The hours and the minutes are given with two digits (the hours or the minutes less than 10 are given with the leading zeroes).

The second line contains integer a ($0 \le a \le 10^4$) — the number of the minutes passed.

Output

The only line should contain the time after a minutes in the format described in the input. Note that you should print exactly two digits for the hours and the minutes (add leading zeroes to the numbers if needed).

See the examples to check the input/output format.

Examples

input

```
23:59
10
```

output

```
00:09
```

input

```
20:20
121
```

output

```
22:21
```

input

```
10:10
0
```

output

```
10:10
```

C. Not Equal on a Segment

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given array a with n integers and m queries. The i-th query is given with three integers l_i, r_i, x_i.

For the i-th query find any position p_i ($l_i \le p_i \le r_i$) so that $a_{p_i} \ne x_i$.

Input

The first line contains two integers n, m ($1 \le n, m \le 2 \cdot 10^5$) — the number of elements in a and the number of queries.

The second line contains n integers a_i ($1 \le a_i \le 10^6$) — the elements of the array a.

Each of the next m lines contains three integers l_i, r_i, x_i ($1 \le l_i \le r_i \le n, 1 \le x_i \le 10^6$) — the parameters of the i-th query.

Output

Print m lines. On the i-th line print integer p_i — the position of any number not equal to x_i in segment $[l_i, r_i]$ or the value -1 if there is no such number.

Examples

input

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

output

```
2
6
-1
4
```

D. Optimal Number Permutation

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have array a that contains all integers from 1 to n twice. You can arbitrary permute any numbers in a.

Let number i be in positions x_i, y_i ($x_i < y_i$) in the permuted array a. Let's define the value $d_i = y_i - x_i$ — the distance between the positions of the number i. Permute the numbers in array a to minimize the value of the sum $s = \sum_{i=1}^{n} (n - i) \cdot |d_i + i - n|$.

Input

The only line contains integer n ($1 \le n \le 5 \cdot 10^5$).

Output

Print $2n$ integers — the permuted array a that minimizes the value of the sum s.

Examples

input

```
2
```

output

```
1 1 2 2
```

input

```
1
```

output

```
1 1
```

E. Ants in Leaves

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Tree is a connected graph without cycles. A leaf of a tree is any vertex connected with exactly one other vertex.

You are given a tree with n vertices and a root in the vertex 1. There is an ant in each leaf of the tree. In one second some ants can simultaneously go to the parent vertex from the vertex they were in. No two ants can be in the same vertex simultaneously except for the root of the tree.

Find the minimal time required for all ants to be in the root of the tree. Note that at start the ants are only in the leaves of the tree.

Input

The first line contains integer n ($2 \le n \le 5 \cdot 10^5$) — the number of vertices in the tree.

Each of the next $n - 1$ lines contains two integers x_i, y_i ($1 \le x_i, y_i \le n$) — the ends of the i-th edge. It is guaranteed that you are given the correct undirected tree.

Output

Print the only integer t — the minimal time required for all ants to be in the root of the tree.

Examples

input

```
12
1 2
1 3
1 4
2 5
2 6
3 7
3 8
3 9
8 10
8 11
8 12
```

output

```
6
```

input

```
2
2 1
```

output

```
1
```

F. The Sum of the k-th Powers

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

There are well-known formulas: $\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$, $\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \ldots + n^2 = \frac{n(2n+1)(n+1)}{6}$, $\sum_{i=1}^{n} i^3 = 1^3 + 2^3 + \ldots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$. Also mathematicians found similar formulas for higher degrees.

Find the value of the sum $\sum_{i=1}^{n} i^k = 1^k + 2^k + \ldots + n^k$ modulo $10^9 + 7$ (so you should find the remainder after dividing the answer by the value $10^9 + 7$).

Input

The only line contains two integers n, k ($1 \le n \le 10^9, 0 \le k \le 10^6$).

Output

Print the only integer a — the remainder after dividing the value of the sum by the value $10^9 + 7$.

Examples

input

```
4 1
```

output

```
10
```

input

```
4 2
```

output

```
30
```

input

```
4 3
```

output

```
100
```

input

```
4 0
```

output

```
4
```