Educational Codeforces Round 10

A. Gabriel and Caterpillar

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

The 9-th grade student Gabriel noticed a caterpillar on a tree when walking around in a forest after the classes. The caterpillar was on the height h_1 cm from the ground. On the height h_2 cm ($h_2 > h_1$) on the same tree hung an apple and the caterpillar was crawling to the apple.

Gabriel is interested when the caterpillar gets the apple. He noted that the caterpillar goes up by a cm per hour by day and slips down by b cm per hour by night.

In how many days Gabriel should return to the forest to see the caterpillar get the apple. You can consider that the day starts at 10 am and finishes at 10 pm. Gabriel's classes finish at 2 pm. You can consider that Gabriel noticed the caterpillar just after the classes at 2 pm.

Note that the forest is magic so the caterpillar can slip down under the ground and then lift to the apple.

Input

The first line contains two integers h_1, h_2 $(1 \le h_1 < h_2 \le 10^5)$ — the heights of the position of the caterpillar and the apple in centimeters.

The second line contains two integers a, b $(1 \le a, b \le 10^5)$ — the distance the caterpillar goes up by day and slips down by night, in centimeters per hour.

Output

Print the only integer k — the number of days Gabriel should wait to return to the forest and see the caterpillar getting the apple.

If the caterpillar can't get the apple print the only integer -1.

Examples

input
```
10 30
2 1
```
output
```
1
```

input
```
10 13
1 1
```
output
```
0
```

input
```
10 19
1 2
```
output
```
-1
```

input
```
1 50
5 4
```
output
```
1
```

Note

In the first example at 10 pm of the first day the caterpillar gets the height 26. At 10 am of the next day it slips down to the height 14. And finally at 6 pm of the same day the caterpillar gets the apple.

Note that in the last example the caterpillar was slipping down under the ground and getting the apple on the next day.

B. z-sort

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

A student of z-school found a kind of sorting called z-sort. The array a with n elements are z-sorted if two conditions hold:

1. $a_i \geq a_{i-1}$ for all even i,
2. $a_i \leq a_{i-1}$ for all odd $i > 1$.

For example the arrays `[1,2,1,2]` and `[1,1,1,1]` are z-sorted while the array `[1,2,3,4]` isn't z-sorted.

Can you make the array z-sorted?

Input

The first line contains a single integer n $(1 \leq n \leq 1000)$ — the number of elements in the array a.

The second line contains n integers a_i $(1 \leq a_i \leq 10^9)$ — the elements of the array a.

Output

If it's possible to make the array a z-sorted print n space separated integers a_i — the elements after z-sort. Otherwise print the only word "`Impossible`".

Examples

input
```
4
1 2 2 1
```
output
```
1 2 1 2
```

input
```
5
1 3 2 2 5
```
output
```
1 5 2 3 2
```

C. Foe Pairs

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

You are given a permutation p of length n. Also you are given m foe pairs (a_i, b_i) $(1 \le a_i, b_i \le n, a_i \ne b_i)$.

Your task is to count the number of different intervals (x, y) $(1 \le x \le y \le n)$ that do not contain any foe pairs. So you shouldn't count intervals (x, y) that contain at least one foe pair in it (the positions and order of the values from the foe pair are not important).

Consider some example: $p = [1, 3, 2, 4]$ and foe pairs are $\{(3, 2), (4, 2)\}$. The interval $(1, 3)$ is incorrect because it contains a foe pair $(3, 2)$. The interval $(1, 4)$ is also incorrect because it contains two foe pairs $(3, 2)$ and $(4, 2)$. But the interval $(1, 2)$ is correct because it doesn't contain any foe pair.

Input

The first line contains two integers n and m $(1 \le n, m \le 3 \cdot 10^5)$ — the length of the permutation p and the number of foe pairs.

The second line contains n distinct integers p_i $(1 \le p_i \le n)$ — the elements of the permutation p.

Each of the next m lines contains two integers (a_i, b_i) $(1 \le a_i, b_i \le n, a_i \ne b_i)$ — the i-th foe pair. Note a foe pair can appear multiple times in the given list.

Output

Print the only integer c — the number of different intervals (x, y) that does not contain any foe pairs.

Note that the answer can be 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.

Examples

input

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

output

```
5
```

input

```
9 5
9 7 2 3 1 4 6 5 8
1 6
4 5
2 7
7 2
2 7
```

output

```
20
```

Note

In the first example the intervals from the answer are $(1, 1)$, $(1, 2)$, $(2, 2)$, $(3, 3)$ and $(4, 4)$.

D. Nested Segments

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given n segments on a line. There are no ends of some segments that coincide. For each segment find the number of segments it contains.

Input

The first line contains a single integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of segments on a line.

Each of the next n lines contains two integers l_i and r_i $(-10^9 \le l_i < r_i \le 10^9)$ — the coordinates of the left and the right ends of the i-th segment. It is guaranteed that there are no ends of some segments that coincide.

Output

Print n lines. The j-th of them should contain the only integer a_j — the number of segments contained in the j-th segment.

Examples

input

```
4
1 8
2 3
4 7
5 6
```

output

```
3
0
1
0
```

input

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

output

```
0
1
1
```

E. Pursuit For Artifacts

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Johnny is playing a well-known computer game. The game are in some country, where the player can freely travel, pass quests and gain an experience.

In that country there are n islands and m bridges between them, so you can travel from any island to any other. In the middle of some bridges are lying ancient powerful artifacts. Johnny is not interested in artifacts, but he can get some money by selling some artifact.

At the start Johnny is in the island a and the artifact-dealer is in the island b (possibly they are on the same island). Johnny wants to find some artifact, come to the dealer and sell it. The only difficulty is that bridges are too old and destroying right after passing over them. Johnnie's character can't swim, fly and teleport, so the problem became too difficult.

Note that Johnny can't pass the half of the bridge, collect the artifact and return to the same island.

Determine if Johnny can find some artifact and sell it.

Input

The first line contains two integers n and m ($1 \le n \le 3 \cdot 10^5$, $0 \le m \le 3 \cdot 10^5$) — the number of islands and bridges in the game.

Each of the next m lines contains the description of the bridge — three integers x_i, y_i, z_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$, $0 \le z_i \le 1$), where x_i and y_i are the islands connected by the i-th bridge, z_i equals to one if that bridge contains an artifact and to zero otherwise. There are no more than one bridge between any pair of islands. It is guaranteed that it's possible to travel between any pair of islands.

The last line contains two integers a and b ($1 \le a, b \le n$) — the islands where are Johnny and the artifact-dealer respectively.

Output

If Johnny can find some artifact and sell it print the only word "YES" (without quotes). Otherwise print the word "NO" (without quotes).

Examples

input

```
6 7
1 2 0
2 3 0
3 1 0
3 4 1
4 5 0
5 6 0
6 4 0
1 6
```

output

```
YES
```

input

```
5 4
1 2 0
2 3 0
3 4 0
2 5 1
1 4
```

output

```
NO
```

input

```
5 6
1 2 0
2 3 0
3 1 0
3 4 0
4 5 1
5 3 0
1 2
```

output

```
YES
```

F. Ants on a Circle

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

n ants are on a circle of length m. An ant travels one unit of distance per one unit of time. Initially, the ant number i is located at the position s_i and is facing in the direction d_i (which is either L or R). Positions are numbered in counterclockwise order starting from some point. Positions of the all ants are distinct.

All the ants move simultaneously, and whenever two ants touch, they will both switch their directions. Note that it is possible for an ant to move in some direction for a half of a unit of time and in opposite direction for another half of a unit of time.

Print the positions of the ants after t time units.

Input

The first line contains three integers n, m and t ($2 \le n \le 3 \cdot 10^5, 2 \le m \le 10^9, 0 \le t \le 10^{18}$) — the number of ants, the length of the circle and the number of time units.

Each of the next n lines contains integer s_i and symbol d_i ($1 \le s_i \le m$ and d_i is either L or R) — the position and the direction of the i-th ant at the start. The directions L and R corresponds to the clockwise and counterclockwise directions, respectively.

It is guaranteed that all positions s_i are distinct.

Output

Print n integers x_j — the position of the j-th ant after t units of time. The ants are numbered from 1 to n in order of their appearing in input.

Examples

input

```
2 4 8
1 R
3 L
```

output

```
1 3
```

input

```
4 8 6
6 R
5 L
1 R
8 L
```

output

```
7 4 2 7
```

input

```
4 8 2
1 R
5 L
6 L
8 R
```

output

```
3 3 4 2
```