A. Fight the Monster

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

A monster is attacking the Cyberland!

Master Yang, a braver, is going to beat the monster. Yang and the monster each have 3 attributes: hitpoints (HP), offensive power (ATK) and defensive power (DEF).

During the battle, every second the monster's HP decrease by $max(0, ATK_Y - DEF_M)$, while Yang's HP decreases by $max(0, ATK_M - DEF_Y)$, where index Y denotes Master Yang and index M denotes monster. Both decreases happen simultaneously Once monster's $HP \le 0$ and the same time Master Yang's $HP > 0$, Master Yang wins.

Master Yang can buy attributes from the magic shop of Cyberland: h bitcoins per HP, a bitcoins per ATK, and d bitcoins per DEF.

Now Master Yang wants to know the minimum number of bitcoins he can spend in order to win.

Input

The first line contains three integers HP_Y, ATK_Y, DEF_Y, separated by a space, denoting the initial HP, ATK and DEF of Master Yang.

The second line contains three integers HP_M, ATK_M, DEF_M, separated by a space, denoting the HP, ATK and DEF of the monster.

The third line contains three integers h, a, d, separated by a space, denoting the price of 1 HP, 1 ATK and 1 DEF.

All numbers in input are **integer** and lie between 1 and 100 inclusively.

Output

The only output line should contain an integer, denoting the minimum bitcoins Master Yang should spend in order to win.

Sample test(s)

input

```
1 2 1
1 100 1
1 100 100
```

output

```
99
```

input

```
100 100 100
1 1 1
1 1 1
```

output

```
0
```

Note

For the first sample, prices for ATK and DEF are extremely high. Master Yang can buy 99 HP, then he can beat the monster with 1 HP left.

For the second sample, Master Yang is strong enough to beat the monster, so he doesn't need to buy anything.

B. Strip

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

Alexandra has a paper strip with n numbers on it. Let's call them a_i from left to right.

Now Alexandra wants to split it into some pieces (possibly 1). For each piece of strip, it must satisfy:

- Each piece should contain at least l numbers.
- The difference between the maximal and the minimal number on the piece should be at most s.

Please help Alexandra to find the minimal number of pieces meeting the condition above.

Input

The first line contains three space-separated integers n, s, l ($1 \le n \le 10^5, 0 \le s \le 10^9, 1 \le l \le 10^5$).

The second line contains n integers a_i separated by spaces ($-10^9 \le a_i \le 10^9$).

Output

Output the minimal number of strip pieces.

If there are no ways to split the strip, output -1.

Sample test(s)

input

```
7 2 2
1 3 1 2 4 1 2
```

output

```
3
```

input

```
7 2 2
1 100 1 100 1 100 1
```

output

```
-1
```

Note

For the first sample, we can split the strip into 3 pieces: $[1, 3, 1], [2, 4], [1, 2]$.

For the second sample, we can't let 1 and 100 be on the same piece, so no solution exists.

C. Prefix Product Sequence

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

Consider a sequence $[a_1, a_2, \ldots, a_n]$. Define its prefix product sequence $[a_1 \mod n, (a_1a_2) \mod n, \cdots, (a_1a_2\cdots a_n) \mod n]$.

Now given n, find a permutation of $[1, 2, \ldots, n]$, such that its prefix product sequence is a permutation of $[0, 1, \ldots, n - 1]$.

Input

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

Output

In the first output line, print "`YES`" if such sequence exists, or print "`NO`" if no such sequence exists.

If any solution exists, you should output n more lines. i-th line contains only an integer a_i. The elements of the sequence should be different positive integers no larger than n.

If there are multiple solutions, you are allowed to print any of them.

Sample test(s)

input

```
7
```

output

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

input

```
6
```

output

```
NO
```

Note

For the second sample, there are no valid sequences.

D. Conveyor Belts

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Automatic Bakery of Cyberland (ABC) recently bought an $n \times m$ rectangle table. To serve the diners, ABC placed seats around the table. The size of each seat is equal to a unit square, so there are $2(n+m)$ seats in total.

ABC placed conveyor belts on each unit square on the table. There are three types of conveyor belts: "^", "<" and ">". A "^" belt can bring things upwards. "<" can bring leftwards and ">" can bring rightwards.

Let's number the rows with 1 to n from top to bottom, the columns with 1 to m from left to right. We consider the seats above and below the top of the table are rows 0 and $n+1$ respectively. Also we define seats to the left of the table and to the right of the table to be column 0 and $m+1$. Due to the conveyor belts direction restriction there are currently no way for a diner sitting in the row $n+1$ to be served.

Given the initial table, there will be q events in order. There are two types of events:

- "A x y" means, a piece of bread will appear at row x and column y (we will denote such position as (x, y)). The bread will follow the conveyor belt, until arriving at a seat of a diner. It is possible that the bread gets stuck in an infinite loop. Your task is to simulate the process, and output the final position of the bread, or determine that there will be an infinite loop.
- "C x y c" means that the type of the conveyor belt at (x, y) is changed to c.

Queries are performed separately meaning that even if the bread got stuck in an infinite loop, it won't affect further queries.

Input

The first line of input contains three integers n, m and q ($1 \le n \le 10^5$, $1 \le m \le 10$, $1 \le q \le 10^5$), separated by a space.

Next n lines, each line contains m characters, describing the table. The characters can only be one of "<^>".

Next q lines, each line describes an event. The format is "`C x y c`" or "`A x y`" (Consecutive elements are separated by a space). It's guaranteed that $1 \le x \le n$, $1 \le y \le m$. c is a character from the set "<^>".

There are at most 10000 queries of "C" type.

Output

For each event of type "A", output two integers tx, ty in a line, separated by a space, denoting the destination of (x, y) is (tx, ty).

If there is an infinite loop, you should output $tx = ty = -1$.

Sample test(s)

input
```
2 2 3
>>
^^
A 2 1
C 1 2 <
A 2 1
```
output
```
1 3
-1 -1
```

input
```
4 5 7
><<^<
^<^^>
>>>^>
>^>>^
A 3 1
A 2 2
C 1 4 <
A 3 1
C 1 2 ^
A 3 1
A 2 2
```
output
```
0 4
-1 -1
-1 -1
0 2
0 2
```

Note

For the first sample:

If the bread goes from $(2, 1)$, it will go out of the table at $(1, 3)$.

After changing the conveyor belt of $(1, 2)$ to "<", when the bread goes from $(2, 1)$ again, it will get stuck at "><", so output is $(-1, -1)$.

E. Tourists

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

There are n cities in Cyberland, numbered from 1 to n, connected by m bidirectional roads. The j-th road connects city a_j and b_j.

For tourists, souvenirs are sold in every city of Cyberland. In particular, city i sell it at a price of w_i.

Now there are q queries for you to handle. There are two types of queries:

- "`C` a w": The price in city a is changed to w.
- "`A` a b": Now a tourist will travel from city a to b. He will choose a route, he also doesn't want to visit a city twice. He will buy souvenirs at the city where the souvenirs are the cheapest (possibly exactly at city a or b). You should output the minimum possible price that he can buy the souvenirs during his travel.

More formally, we can define routes as follow:

- A route is a sequence of cities $[x_1, x_2, ..., x_k]$, where k is a certain positive integer.
- For any $1 \leq i < j \leq k, x_i \neq x_j$.
- For any $1 \leq i < k$, there is a road connecting x_i and x_{i+1}.
- The minimum price of the route is $min(w_{x_1}, w_{x_2}, ..., w_{x_k})$.
- The required answer is the minimum value of the minimum prices of all valid routes from a to b.

Input

The first line of input contains three integers n, m, q $(1 \leq n, m, q \leq 10^5)$, separated by a single space.

Next n lines contain integers w_i $(1 \leq w_i \leq 10^9)$.

Next m lines contain pairs of space-separated integers a_j and b_j $(1 \leq a_j, b_j \leq n, a_j \neq b_j)$.

It is guaranteed that there is at most one road connecting the same pair of cities. There is always at least one valid route between any two cities.

Next q lines each describe a query. The format is "`C` a w" or "`A` a b" $(1 \leq a, b \leq n, 1 \leq w \leq 10^9)$.

Output

For each query of type "`A`", output the corresponding answer.

Sample test(s)

input

```
3 3 3
1
2
3
1 2
2 3
1 3
A 2 3
C 1 5
A 2 3
```

output

```
1
2
```

input

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

```
output
```

```
2
1
5
3
```

Note

For the second sample, an optimal routes are:

From 2 to 3 it is $[2, 3]$.

From 6 to 4 it is $[6, 5, 1, 2, 4]$.

From 6 to 7 it is $[6, 5, 7]$.

From 3 to 3 it is $[3]$.