A. Appleman and Toastman

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

Appleman and Toastman play a game. Initially Appleman gives one group of n numbers to the Toastman, then they start to complete the following tasks:

- Each time Toastman gets a group of numbers, he sums up all the numbers and adds this sum to the score. Then he gives the group to the Appleman.
- Each time Appleman gets a group consisting of a single number, he throws this group out. Each time Appleman gets a group consisting of more than one number, he splits the group into two non-empty groups (he can do it in any way) and gives each of them to Toastman.

After guys complete all the tasks they look at the score value. What is the maximum possible value of score they can get?

Input

The first line contains a single integer n ($1 \le n \le 3 \cdot 10^5$). The second line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^6$) — the initial group that is given to Toastman.

Output

Print a single integer — the largest possible score.

Sample test(s)

input

```
3
3 1 5
```

output

```
26
```

input

```
1
10
```

output

```
10
```

Note

Consider the following situation in the first example. Initially Toastman gets group [3, 1, 5] and adds 9 to the score, then he give the group to Appleman. Appleman splits group [3, 1, 5] into two groups: [3, 5] and [1]. Both of them should be given to Toastman. When Toastman receives group [1], he adds 1 to score and gives the group to Appleman (he will throw it out). When Toastman receives group [3, 5], he adds 8 to the score and gives the group to Appleman. Appleman splits [3, 5] in the only possible way: [5] and [3]. Then he gives both groups to Toastman. When Toastman receives [5], he adds 5 to the score and gives the group to Appleman (he will throws it out). When Toastman receives [3], he adds 3 to the score and gives the group to Appleman (he will throws it out). Finally Toastman have added 9 + 1 + 8 + 5 + 3 = 26 to the score. This is the optimal sequence of actions.

B. Appleman and Tree

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

Appleman has a tree with n vertices. Some of the vertices (at least one) are colored black and other vertices are colored white.

Consider a set consisting of k $(0 \le k < n)$ edges of Appleman's tree. If Appleman deletes these edges from the tree, then it will split into $(k+1)$ parts. Note, that each part will be a tree with colored vertices.

Now Appleman wonders, what is the number of sets splitting the tree in such a way that each resulting part will have exactly one black vertex? Find this number modulo 1000000007 $(10^9 + 7)$.

Input

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

The second line contains the description of the tree: $n - 1$ integers $p_0, p_1, ..., p_{n-2}$ $(0 \le p_i \le i)$. Where p_i means that there is an edge connecting vertex $(i+1)$ of the tree and vertex p_i. Consider tree vertices are numbered from 0 to $n - 1$.

The third line contains the description of the colors of the vertices: n integers $x_0, x_1, ..., x_{n-1}$ (x_i is either 0 or 1). If x_i is equal to 1, vertex i is colored black. Otherwise, vertex i is colored white.

Output

Output a single integer — the number of ways to split the tree modulo 1000000007 $(10^9 + 7)$.

Sample test(s)

input

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

output

```
2
```

input

```
6
0 1 1 0 4
1 1 0 0 1 0
```

output

```
1
```

input

```
10
0 1 2 1 4 4 4 0 8
0 0 0 1 0 1 1 0 0 1
```

output

```
27
```

C. Appleman and a Sheet of Paper

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

Appleman has a very big sheet of paper. This sheet has a form of rectangle with dimensions $1 \times n$. Your task is help Appleman with folding of such a sheet. Actually, you need to perform q queries. Each query will have one of the following types:

1. Fold the sheet of paper at position p_i. After this query the leftmost part of the paper with dimensions $1 \times p_i$ must be above the rightmost part of the paper with dimensions $1 \times ([current\ width\ of\ sheet] - p_i)$.
2. Count what is the total width of the paper pieces, if we will make two described later cuts and consider only the pieces between the cuts. We will make one cut at distance l_i from the left border of the current sheet of paper and the other at distance r_i from the left border of the current sheet of paper.

Please look at the explanation of the first test example for better understanding of the problem.

Input

The first line contains two integers: n and q ($1 \le n \le 10^5$; $1 \le q \le 10^5$) — the width of the paper and the number of queries.

Each of the following q lines contains one of the described queries in the following format:

- "$1\ p_i$" ($1 \le p_i < [current\ width\ of\ sheet]$) — the first type query.
- "$2\ l_i\ r_i$" ($0 \le l_i < r_i \le [current\ width\ of\ sheet]$) — the second type query.

Output

For each query of the second type, output the answer.

Sample test(s)

input

```
7 4
1 3
1 2
2 0 1
2 1 2
```

output

```
4
3
```

input

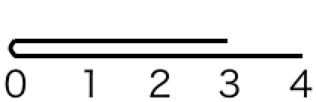
```
10 9
2 2 9
1 1
2 0 1
1 8
2 0 8
1 2
2 1 3
1 4
2 2 4
```

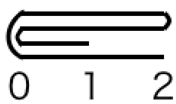
output

```
7
2
10
4
5
```

Note

The pictures below show the shapes of the paper during the queries of the first example:





After the first fold operation the sheet has width equal to 4, after the second one the width of the sheet equals to 2.

D. Appleman and Complicated Task

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

Toastman came up with a very complicated task. He gives it to Appleman, but Appleman doesn't know how to solve it. Can you help him?

Given a $n \times n$ checkerboard. Each cell of the board has either character '$\texttt{x}$', or character '$\texttt{o}$', or nothing. How many ways to fill all the empty cells with '$\texttt{x}$' or '$\texttt{o}$' (each cell must contain only one character in the end) are there, such that for each cell the number of adjacent cells with '$\texttt{o}$' will be even? Find the number of ways modulo 1000000007 $(10^9 + 7)$. Two cells of the board are adjacent if they share a side.

Input

The first line contains two integers n, k $(1 \le n, k \le 10^5)$ — the size of the board, and the number of cells that has characters initially.

Then k lines follows. The i-th line contains two integers and a character: a_i, b_i, c_i $(1 \le a_i, b_i \le n$; c_i is either '$\texttt{o}$' or '$\texttt{x}$'). This line means: there is a character c_i in the cell that is located on the intersection of the a_i-th row and b_i-th column. All the given cells are distinct.

Consider that the rows are numbered from 1 to n from top to bottom. Analogically, the columns are numbered from 1 to n from left to right.

Output

Print a single integer — the answer to the problem.

Sample test(s)

input

```
3 2
1 1 x
2 2 o
```

output

```
2
```

input

```
4 3
2 4 x
3 4 x
3 2 x
```

output

```
2
```

Note

In the first example there are two ways:

```
xxo    xoo
xox    ooo
oxx    oox
```

E. Appleman and a Game

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

Appleman and Toastman like games. Today they play a game with strings with the following rules. Firstly Toastman tells Appleman two strings s and t both consisting only of letters 'A', 'B', 'C', 'D'. Then Appleman must build string s as quickly as possible. Initially he has empty string, and in one second he can append to end of the current string any contiguous substring of t.

Now, Toastman and Appleman are beginning to play the game. Toastman has already told string t to Appleman, but he hasn't come up with string s yet. Toastman only thinks, that he should choose string s consisting of n characters. Of course, he wants to find the worst string for Appleman (such string, that Appleman will spend as much time as possible during the game). Tell Toastman, how much time will Appleman spend during the game if Toastman finds the worst string for him. You can assume that Appleman plays optimally, therefore he builds any string s in minimal possible time.

Input

The first line contains an integer n $(1 \le n \le 10^{18})$. The second line contains string t $(1 \le |t| \le 10^5)$. String t consists of only letters 'A', 'B', 'C', 'D'. Each letter appears at least once in string t.

Output

Print a single integer — the largest possible time Appleman needs.

Sample test(s)

input
```
5
ABCCAD
```
output
```
5
```

input
```
5
AAABACADBABBBCBDCACBCCCDDDBDCDD
```
output
```
4
```

Note

In the first example, Toastman can choose s equal to "AAAAA".

In the second example, Toastman can choose s equal to "DADDA".