A. Kefa and First Steps

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Kefa decided to make some money doing business on the Internet for exactly n days. He knows that on the i-th day ($1 \le i \le n$) he makes a_i money. Kefa loves progress, that's why he wants to know the length of the maximum non-decreasing subsegment in sequence a_i. Let us remind you that the subsegment of the sequence is its continuous fragment. A subsegment of numbers is called non-decreasing if all numbers in it follow in the non-decreasing order.

Help Kefa cope with this task!

Input

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

The second line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

Output

Print a single integer — the length of the maximum non-decreasing subsegment of sequence a.

Sample test(s)

input

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

output

```
3
```

input

```
3
2 2 9
```

output

```
3
```

Note

In the first test the maximum non-decreasing subsegment is the numbers from the third to the fifth one.

In the second test the maximum non-decreasing subsegment is the numbers from the first to the third one.

B. Kefa and Company

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

Kefa wants to celebrate his first big salary by going to restaurant. However, he needs company.

Kefa has n friends, each friend will agree to go to the restaurant if Kefa asks. Each friend is characterized by the amount of money he has and the friendship factor in respect to Kefa. The parrot doesn't want any friend to feel poor compared to somebody else in the company (Kefa doesn't count). A friend feels poor if in the company there is someone who has at least d units of money more than he does. Also, Kefa wants the total friendship factor of the members of the company to be maximum. Help him invite an optimal company!

Input

The first line of the input contains two space-separated integers, n and d ($1 \le n \le 10^5$, $\mathbf{1} \le d \le 10^9$) — the number of Kefa's friends and the minimum difference between the amount of money in order to feel poor, respectively.

Next n lines contain the descriptions of Kefa's friends, the $(i+1)$-th line contains the description of the i-th friend of type m_i, s_i ($0 \le m_i, s_i \le 10^9$) — the amount of money and the friendship factor, respectively.

Output

Print the maximum total friendship factir that can be reached.

Sample test(s)

input

```
4 5
75 5
0 100
150 20
75 1
```

output

```
100
```

input

```
5 100
0 7
11 32
99 10
46 8
87 54
```

output

```
111
```

Note

In the first sample test the most profitable strategy is to form a company from only the second friend. At all other variants the total degree of friendship will be worse.

In the second sample test we can take all the friends.

C. Kefa and Park

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

Kefa decided to celebrate his first big salary by going to the restaurant.

He lives by an unusual park. The park is a rooted tree consisting of n vertices with the root at vertex 1. Vertex 1 also contains Kefa's house. Unfortunaely for our hero, the park also contains cats. Kefa has already found out what are the vertices with cats in them.

The leaf vertices of the park contain restaurants. Kefa wants to choose a restaurant where he will go, but unfortunately he is very afraid of cats, so there is no way he will go to the restaurant if the path from the restaurant to his house contains more than m **consecutive** vertices with cats.

Your task is to help Kefa count the number of restaurants where he can go.

Input

The first line contains two integers, n and m ($2 \le n \le 10^5$, $1 \le m \le n$) — the number of vertices of the tree and the maximum number of consecutive vertices with cats that is still ok for Kefa.

The second line contains n integers $a_1, a_2, ..., a_n$, where each a_i either equals to 0 (then vertex i has no cat), or equals to 1 (then vertex i has a cat).

Next n - 1 lines contains the edges of the tree in the format "x_i y_i" (without the quotes) ($1 \le x_i, y_i \le n$, $x_i \ne y_i$), where x_i and y_i are the vertices of the tree, connected by an edge.

It is guaranteed that the given set of edges specifies a tree.

Output

A single integer — the number of distinct leaves of a tree the path to which from Kefa's home contains at most m consecutive vertices with cats.

Sample test(s)

input

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

output

```
2
```

input

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

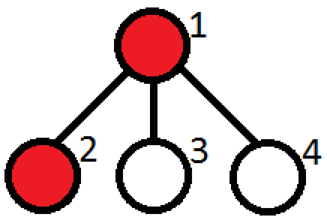
output

```
2
```

Note

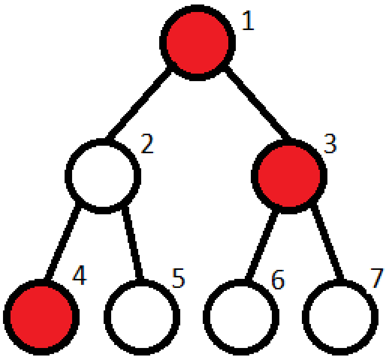
Let us remind you that a *tree* is a connected graph on n vertices and n - 1 edge. A *rooted* tree is a tree with a special vertex called *root*. In a rooted tree among any two vertices connected by an edge, one vertex is a parent (the one closer to the root), and the other one is a child. A vertex is called a *leaf*, if it has no children.

Note to the first sample test:



The vertices containing cats are marked red. The restaurants are at vertices 2, 3, 4. Kefa can't go only to the restaurant located at vertex 2.

Note to the second sample test:



The restaurants are located at vertices 4, 5, 6, 7. Kefa can't go to restaurants 6, 7.

D. Kefa and Dishes

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

When Kefa came to the restaurant and sat at a table, the waiter immediately brought him the menu. There were n dishes. Kefa knows that he needs exactly m dishes. But at that, he doesn't want to order the same dish twice to taste as many dishes as possible.

Kefa knows that the i-th dish gives him a_i units of satisfaction. But some dishes do not go well together and some dishes go very well together. Kefa set to himself k rules of eating food of the following type — if he eats dish x exactly before dish y (there should be no other dishes between x and y), then his satisfaction level raises by c.

Of course, our parrot wants to get some maximal possible satisfaction from going to the restaurant. Help him in this hard task!

Input

The first line of the input contains three space-separated numbers, n, m and k $(1 \le m \le n \le 18, 0 \le k \le n * (n - 1))$ — the number of dishes on the menu, the number of portions Kefa needs to eat to get full and the number of eating rules.

The second line contains n space-separated numbers a_i, $(0 \le a_i \le 10^9)$ — the satisfaction he gets from the i-th dish.

Next k lines contain the rules. The i-th rule is described by the three numbers x_i, y_i and c_i $(1 \le x_i, y_i \le n, 0 \le c_i \le 10^9)$. That means that if you eat dish x_i right before dish y_i, then the Kefa's satisfaction increases by c_i. It is guaranteed that there are no such pairs of indexes i and j $(1 \le i < j \le k)$, that $x_i = x_j$ and $y_i = y_j$.

Output

In the single line of the output print the maximum satisfaction that Kefa can get from going to the restaurant.

Sample test(s)

input

```
2 2 1
1 1
2 1 1
```

output

```
3
```

input

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

output

```
12
```

Note

In the first sample it is best to first eat the second dish, then the first one. Then we get one unit of satisfaction for each dish and plus one more for the rule.

In the second test the fitting sequences of choice are 4 2 1 or 2 1 4. In both cases we get satisfaction 7 for dishes and also, if we fulfill rule 1, we get an additional satisfaction 5.

E. Kefa and Watch

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

One day Kefa the parrot was walking down the street as he was on the way home from the restaurant when he saw something glittering by the road. As he came nearer he understood that it was a watch. He decided to take it to the pawnbroker to earn some money.

The pawnbroker said that each watch contains a serial number represented by a string of digits from 0 to 9, and the more quality checks this number passes, the higher is the value of the watch. The check is defined by three positive integers l, r and d. The watches pass a check if a substring of the serial number from l to r has period d. Sometimes the pawnbroker gets distracted and Kefa changes in some substring of the serial number all digits to c in order to increase profit from the watch.

The seller has a lot of things to do to begin with and with Kefa messing about, he gave you a task: to write a program that determines the value of the watch.

Let us remind you that number x is called a period of string s ($1 \le x \le |s|$), if $s_i = s_{i+x}$ for all i from 1 to $|s| - x$.

Input

The first line of the input contains three positive integers n, m and k ($1 \le n \le 10^5$, $1 \le m + k \le 10^5$) — the length of the serial number, the number of change made by Kefa and the number of quality checks.

The second line contains a serial number consisting of n digits.

Then $m + k$ lines follow, containing either checks or changes.

The changes are given as 1 l r c ($1 \le l \le r \le n$, $0 \le c \le 9$). That means that Kefa changed all the digits from the l-th to the r-th to be c.

The checks are given as 2 l r d ($1 \le l \le r \le n$, $1 \le d \le r - l + 1$).

Output

For each check on a single line print "YES" if the watch passed it, otherwise print "NO".

Sample test(s)

input

```
3 1 2
112
2 2 3 1
1 1 3 8
2 1 2 1
```

output

```
NO
YES
```

input

```
6 2 3
334934
2 2 5 2
1 4 4 3
2 1 6 3
1 2 3 8
2 3 6 1
```

output

```
NO
YES
NO
```

Note

In the first sample test two checks will be made. In the first one substring "12" is checked on whether or not it has period 1, so the answer is "NO". In the second one substring "88", is checked on whether or not it has period 1, and it has this period, so the answer is "YES".

In the second statement test three checks will be made. The first check processes substring "3493", which doesn't have period 2. Before the second check the string looks as "334334", so the answer to it is "YES". And finally, the third check processes substring "8334", which does not have period 1.