A. Stones on the Table

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

There are n stones on the table in a row, each of them can be red, green or blue. Count the minimum number of stones to take from the table so that any two neighboring stones had different colors. Stones in a row are considered neighboring if there are no other stones between them.

Input

The first line contains integer n $(1 \le n \le 50)$ — the number of stones on the table.

The next line contains string s, which represents the colors of the stones. We'll consider the stones in the row numbered from 1 to n from left to right. Then the i-th character s equals "R", if the i-th stone is red, "G", if it's green and "B", if it's blue.

Output

Print a single integer — the answer to the problem.

Examples

input

```
3
RRG
```

output

```
1
```

input

```
5
RRRRR
```

output

```
4
```

input

```
4
BRBG
```

output

```
0
```

B. Queue at the School

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

During the break the schoolchildren, boys and girls, formed a queue of n people in the canteen. Initially the children stood in the order they entered the canteen. However, after a while the boys started feeling awkward for standing in front of the girls in the queue and they started letting the girls move forward each second.

Let's describe the process more precisely. Let's say that the positions in the queue are sequentially numbered by integers from 1 to n, at that the person in the position number 1 is served first. Then, if at time x a boy stands on the i-th position and a girl stands on the $(i+1)$-th position, then at time $x+1$ the i-th position will have a girl and the $(i+1)$-th position will have a boy. The time is given in seconds.

You've got the initial position of the children, at the initial moment of time. Determine the way the queue is going to look after t seconds.

Input

The first line contains two integers n and t $(1 \le n, t \le 50)$, which represent the number of children in the queue and the time after which the queue will transform into the arrangement you need to find.

The next line contains string s, which represents the schoolchildren's initial arrangement. If the i-th position in the queue contains a boy, then the i-th character of string s equals "B", otherwise the i-th character equals "G".

Output

Print string a, which describes the arrangement after t seconds. If the i-th position has a boy after the needed time, then the i-th character a must equal "B", otherwise it must equal "G".

Examples

input

```
5 1
BGGBG
```

output

```
GBGGB
```

input

```
5 2
BGGBG
```

output

```
GGBGB
```

input

```
4 1
GGGB
```

output

```
GGGB
```

C. Below the Diagonal

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

You are given a square matrix consisting of n rows and n columns. We assume that the rows are numbered from 1 to n from top to bottom and the columns are numbered from 1 to n from left to right. Some cells ($n - 1$ cells in total) of the the matrix are filled with ones, the remaining cells are filled with zeros. We can apply the following operations to the matrix:

1. Swap i-th and j-th rows of the matrix;
2. Swap i-th and j-th columns of the matrix.

You are asked to transform the matrix into a special form using these operations. In that special form all the ones must be in the cells that lie below the main diagonal. Cell of the matrix, which is located on the intersection of the i-th row and of the j-th column, lies below the main diagonal if $i > j$.

Input

The first line contains an integer n $(2 \le n \le 1000)$ — the number of rows and columns. Then follow $n - 1$ lines that contain one's positions, one per line. Each position is described by two integers x_k, y_k $(1 \le x_k, y_k \le n)$, separated by a space. A pair (x_k, y_k) means that the cell, which is located on the intersection of the x_k-th row and of the y_k-th column, contains one.

It is guaranteed that all positions are distinct.

Output

Print the description of your actions. These actions should transform the matrix to the described special form.

In the first line you should print a non-negative integer m $(m \le 10^5)$ — the number of actions. In each of the next m lines print three space-separated integers t, i, j $(1 \le t \le 2, 1 \le i, j \le n, i \ne j)$, where $t = 1$ if you want to swap rows, $t = 2$ if you want to swap columns, and i and j denote the numbers of rows or columns respectively.

Please note, that you do not need to minimize the number of operations, but their number should not exceed 10^5. If there are several solutions, you may print any of them.

Examples

input

```
2
1 2
```

output

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

input

```
3
3 1
1 3
```

output

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

input

```
3
2 1
3 2
```

output

```
0
```

D. BerDonalds

time limit per test: 5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

BerDonalds, a well-known fast food restaurant, is going to open a cafe in Bertown. The important thing is to choose the new restaurant's location so that it would be easy to get there. The Bertown road system is represented by n junctions, connected by m bidirectional roads. For each road we know its length. We also know that we can get from any junction to any other one, moving along the roads.

Your task is to find such location of the restaurant, that the shortest distance along the roads from the cafe to the farthest junction would be minimum. Note that the restaurant can be located not only on the junction, but at any point of any road.

Input

The first line contains two integers n and m ($2 \le n \le 200;\ n-1 \le m \le \frac{n(n-1)}{2}$) — the number of junctions and the number of roads, correspondingly. Then m lines follow, describing all Bertown roads. Each road is described by three integers a_i, b_i, w_i ($1 \le a_i, b_i \le n, a_i \ne b_i;\ 1 \le w_i \le 10^5$), where a_i and b_i are the numbers of the junctions, connected by the i-th road, and w_i is the length of the i-th road.

It is guaranteed that each road connects two distinct junctions, there is at most one road between any two junctions, and you can get from any junction to any other one.

Output

Print a single real number — the shortest distance from the optimal restaurant location to the farthest junction. The answer will be considered correct, if its absolute or relative error doesn't exceed 10^{-9}.

Examples

input

```
2 1
1 2 1
```

output

```
0.50
```

input

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

output

```
1.00
```

input

```
3 2
1 2 100
2 3 1
```

output

```
50.50
```

E. More Queries to Array...

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You've got an array, consisting of n integers: $a_1, a_2, ..., a_n$. Your task is to quickly run the queries of two types:

1. Assign value x to all elements from l to r inclusive. After such query the values of the elements of array $a_l, a_{l+1}, ..., a_r$ become equal to x.
2. Calculate and print sum $\sum_{i=l}^{r} a_i \cdot (i-l+1)^k$, where k doesn't exceed 5. As the value of the sum can be rather large, you should print it modulo 1000000007 $(10^9 + 7)$.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^5$), showing, how many numbers are in the array and the number of queries, correspondingly. The second line contains n integers: $a_1, a_2, ..., a_n$ ($0 \le a_i \le 10^9$) — the initial values of the array elements.

Then m queries follow, one per line:

1. The assign query has the following format: "$= \ l \ r \ x$", ($1 \le l \le r \le n$; $0 \le x \le 10^9$).
2. The query to calculate the sum has the following format: "$? \ l \ r \ k$", ($1 \le l \le r \le n$; $0 \le k \le 5$).

All numbers in the input are integers.

Output

For each query to calculate the sum print an integer — the required sum modulo 1000000007 $(10^9 + 7)$.

Examples

input

```
4 5
5 10 2 1
? 1 2 1
= 2 2 0
? 2 4 3
= 1 4 1
? 1 4 5
```

output

```
25
43
1300
```

input

```
3 1
1000000000 1000000000 1000000000
? 1 3 0
```

output

```
999999986
```