Codeforces Round #295 (Div. 1)

A. DNA Alignment

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

Vasya became interested in bioinformatics. He's going to write an article about similar cyclic DNA sequences, so he invented a new method for determining the similarity of cyclic sequences.

Let's assume that strings s and t have the same length n, then the function $h(s, t)$ is defined as the number of positions in which the respective symbols of s and t are *the same*. Function $h(s, t)$ can be used to define the function of Vasya distance $\rho(s, t)$:

$$\rho(s,t) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} h(\text{shift}(s,i), \text{shift}(t,j)),$$

where $\text{shift}(s, i)$ is obtained from string s, by applying left circular shift i times. For example,

$$\rho("AGC", "CGT") =$$
$$h("AGC", "CGT") + h("AGC", "GTC") + h("AGC", "TCG") +$$
$$h("GCA", "CGT") + h("GCA", "GTC") + h("GCA", "TCG") +$$
$$h("CAG", "CGT") + h("CAG", "GTC") + h("CAG", "TCG") =$$
$$1 + 1 + 0 + 0 + 1 + 1 + 1 + 0 + 1 = 6$$

Vasya found a string s of length n on the Internet. Now he wants to count how many strings t there are such that the Vasya distance from the string s attains maximum possible value. Formally speaking, t must satisfy the equation: $\rho(s,t) = \max_{u:|u|=|s|} \rho(s,u)$.

Vasya could not try all possible strings to find an answer, so he needs your help. As the answer may be very large, count the number of such strings modulo $10^9 + 7$.

Input

The first line of the input contains a single integer n ($1 \le n \le 10^5$).

The second line of the input contains a single string of length n, consisting of characters "ACGT".

Output

Print a single number — the answer modulo $10^9 + 7$.

Sample test(s)

input
```
1
C
```
output
```
1
```

input
```
2
AG
```
output
```
4
```

input
```
3
TTT
```
output
```
1
```

Note

Please note that if for two distinct strings t_1 and t_2 values $\rho(s, t_1)$ и $\rho(s, t_2)$ are maximum among all possible t, then both strings must be taken into account in the answer even if one of them can be obtained by a circular shift of another one.

In the first sample, there is $\rho("C", "C") = 1$, for the remaining strings t of length 1 the value of $\rho(s, t)$ is 0.

In the second sample, $\rho("AG", "AG") = \rho("AG", "GA") = \rho("AG", "AA") = \rho("AG", "GG") = 4$.

In the third sample, $\rho("TTT", "TTT") = 27$

B. Cubes

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

Once Vasya and Petya assembled a figure of m cubes, each of them is associated with a number between 0 and $m - 1$ (inclusive, each number appeared exactly once). Let's consider a coordinate system such that the OX is the ground, and the OY is directed upwards. Each cube is associated with the coordinates of its lower left corner, these coordinates are integers for each cube.

The figure turned out to be *stable*. This means that for any cube that is not on the ground, there is at least one cube under it such that those two cubes touch **by a side or a corner**. More formally, this means that for the cube with coordinates (x, y) either $y = 0$, or there is a cube with coordinates $(x - 1, y - 1)$, $(x, y - 1)$ or $(x + 1, y - 1)$.

Now the boys want to disassemble the figure and put all the cubes in a row. In one step the cube is removed from the figure and being put to the right of the blocks that have already been laid. The guys remove the cubes in such order that the figure remains stable. To make the process more interesting, the guys decided to play the following game. The guys take out the cubes from the figure in turns. It is easy to see that after the figure is disassembled, the integers written on the cubes form a number, written in the m-ary positional numerical system (possibly, with a leading zero). Vasya wants the resulting number to be maximum possible, and Petya, on the contrary, tries to make it as small as possible. Vasya starts the game.

Your task is to determine what number is formed after the figure is disassembled, if the boys play optimally. Determine the remainder of the answer modulo $10^9 + 9$.

Input

The first line contains number m ($2 \le m \le 10^5$).

The following m lines contain the coordinates of the cubes x_i, y_i ($-10^9 \le x_i \le 10^9$, $0 \le y_i \le 10^9$) in ascending order of numbers written on them. It is guaranteed that the original figure is stable.

No two cubes occupy the same place.

Output

In the only line print the answer to the problem.

Sample test(s)

input
```
3
2 1
1 0
0 1
```
output
```
19
```

input
```
5
0 0
0 1
0 2
0 3
0 4
```
output
```
2930
```

C. Pluses everywhere

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

Vasya is sitting on an extremely boring math class. To have fun, he took a piece of paper and wrote out n numbers on a single line. After that, Vasya began to write out different ways to put pluses ("`+`") in the line between certain digits in the line so that the result was a correct arithmetic expression; formally, no two pluses in such a partition can stand together (between any two adjacent pluses there must be at least one digit), and no plus can stand at the beginning or the end of a line. For example, in the string `100500`, ways `100500` (add no pluses), `1+00+500` or `10050+0` are correct, and ways `100++500`, `+1+0+0+5+0+0` or `100500+` are incorrect.

The lesson was long, and Vasya has written all the correct ways to place exactly k pluses in a string of digits. At this point, he got caught having fun by a teacher and he was given the task to calculate the sum of all the resulting arithmetic expressions by the end of the lesson (when calculating the value of an expression the leading zeros should be ignored). As the answer can be large, Vasya is allowed to get only its remainder modulo $10^9 + 7$. Help him!

Input

The first line contains two integers, n and k ($0 \le k < n \le 10^5$).

The second line contains a string consisting of n digits.

Output

Print the answer to the problem modulo $10^9 + 7$.

Sample test(s)

input

```
3 1
108
```

output

```
27
```

input

```
3 2
108
```

output

```
9
```

Note

In the first sample the result equals $(1 + 08) + (10 + 8) = 27$.

In the second sample the result equals $1 + 0 + 8 = 9$.

D. Shop

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

Vasya plays one very well-known and extremely popular MMORPG game. His game character has k skill; currently the i-th of them equals to a_i. Also this game has a common rating table in which the participants are ranked according to the **product** of all the skills of a hero in the descending order.

Vasya decided to 'upgrade' his character via the game store. This store offers n possible ways to improve the hero's skills; Each of these ways belongs to one of three types:

1. assign the i-th skill to b;
2. add b to the i-th skill;
3. multiply the i-th skill by b.

Unfortunately, a) every improvement can only be used once; b) the money on Vasya's card is enough only to purchase not more than m of the n improvements. Help Vasya to reach the highest ranking in the game. To do this tell Vasya which of improvements he has to purchase and in what order he should use them to make his rating become as high as possible. If there are several ways to achieve it, print any of them.

Input

The first line contains three numbers — k, n, m ($1 \le k \le 10^5$, $0 \le m \le n \le 10^5$) — the number of skills, the number of improvements on sale and the number of them Vasya can afford.

The second line contains k space-separated numbers a_i ($1 \le a_i \le 10^6$), the initial values of skills.

Next n lines contain 3 space-separated numbers t_j, i_j, b_j ($1 \le t_j \le 3$, $1 \le i_j \le k$, $1 \le b_j \le 10^6$) — the type of the j-th improvement (1 for assigning, 2 for adding, 3 for multiplying), the skill to which it can be applied and the value of b for this improvement.

Output

The first line should contain a number l ($0 \le l \le m$) — the number of improvements you should use.

The second line should contain l distinct space-separated numbers v_i ($1 \le v_i \le n$) — the indices of improvements in the order in which they should be applied. The improvements are numbered starting from 1, in the order in which they appear in the input.

Sample test(s)

input

```
2 4 3
13 20
1 1 14
1 2 30
2 1 6
3 2 2
```

output

```
3
2 3 4
```

E. Cycling City

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

You are organizing a cycling race on the streets of the city. The city contains n junctions, some pairs of them are connected by roads; on each road you can move in any direction. No two roads connect the same pair of intersections, and no road connects the intersection with itself.

You want the race to be open to both professional athletes and beginner cyclists, and that's why you will organize the race in three nominations: easy, moderate and difficult; each participant will choose the more suitable nomination. For each nomination you must choose the route — the chain of junctions, consecutively connected by roads. Routes must meet the following conditions:

- all three routes should start at the same intersection, and finish at the same intersection (place of start and finish can't be the same);
- to avoid collisions, no two routes can have common junctions (except for the common start and finish), and can not go along the same road (irrespective of the driving direction on the road for those two routes);
- no route must pass twice through the same intersection or visit the same road twice (irrespective of the driving direction on the road for the first and second time of visit).

Preparing for the competition is about to begin, and you need to determine the routes of the race as quickly as possible. The length of the routes is not important, it is only important that all the given requirements were met.

Input

The first line contains two integers n and m ($1 \le n, m \le 2 \cdot 10^5$) — the number of intersections and roads, respectively.

The following m lines contain two integers — the numbers of the intersections connected by a road (the intersections are numbered starting with 1). It is guaranteed that each pair of intersections is connected by no more than one road, and no road connects the intersection to itself.

Please note that it is not guaranteed that you can get from any junction to any other one by using the roads.

Output

If it is possible to create the routes, in the first line print "YES". In the next three lines print the descriptions of each of the three routes in the format "l p_1 ... p_l", where l is the number of intersections in the route, and $p_1, ..., p_l$ are their numbers in the order they follow. The routes must meet all the requirements specified in the statement.

If it is impossible to make the routes in accordance with the requirements, print NO.

Sample test(s)

input

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

output

```
NO
```

input

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

output

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