Russian Code Cup 2016 - Finals [Unofficial Mirror, Div. 1 Only Recommended]

A. Closing ceremony

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

The closing ceremony of Squanch Code Cup is held in the big hall with $n \times m$ seats, arranged in n rows, m seats in a row. Each seat has two coordinates (x, y) $(1 \le x \le n, 1 \le y \le m)$.

There are two queues of people waiting to enter the hall: k people are standing at $(0, 0)$ and $n \cdot m - k$ people are standing at $(0, m + 1)$. Each person should have a ticket for a specific seat. If person p at (x, y) has ticket for seat (x_p, y_p) then he should walk $|x - x_p| + |y - y_p|$ to get to his seat.

Each person has a stamina — the maximum distance, that the person agrees to walk. You should find out if this is possible to distribute all $n \cdot m$ tickets in such a way that each person has enough stamina to get to their seat.

Input

The first line of input contains two integers n and m $(1 \le n \cdot m \le 10^4)$ — the size of the hall.

The second line contains several integers. The first integer k $(0 \le k \le n \cdot m)$ — the number of people at $(0, 0)$. The following k integers indicate stamina of each person there.

The third line also contains several integers. The first integer l $(l = n \cdot m - k)$ — the number of people at $(0, m + 1)$. The following l integers indicate stamina of each person there.

The stamina of the person is a positive integer less that or equal to $n + m$.

Output

If it is possible to distribute tickets between people in the described manner print "YES", otherwise print "NO".

Examples

input

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

output

```
YES
```

input

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

output

```
NO
```

B. Cactusophobia

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

Tree is a connected undirected graph that has no cycles. Edge cactus is a connected undirected graph without loops and parallel edges, such that each edge belongs to at most one cycle.

Vasya has an edge cactus, each edge of this graph has some color.

Vasya would like to remove the minimal number of edges in such way that his cactus turned to a tree. Vasya wants to make it in such a way that there were edges of as many different colors in the resulting tree, as possible. Help him to find how many different colors can the resulting tree have.

Input

The first line contains two integers: n, m $(2 \le n \le 10\,000)$ — the number of vertices and the number of edges in Vasya's graph, respectively.

The following m lines contain three integers each: u, v, c $(1 \le u, v \le n,\ u \ne v,\ 1 \le c \le m)$ — the numbers of vertices connected by the corresponding edge, and its color. It is guaranteed that the described graph is indeed an edge cactus.

Output

Output one integer: the maximal number of different colors that the resulting tree can have.

Examples

input

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

output

```
3
```

input

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

output

```
6
```

C. Homework

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

Today Peter has got an additional homework for tomorrow. The teacher has given three integers to him: n, m and k, and asked him to mark one or more squares on a square grid of size $n \times m$.

The marked squares must form a connected figure, and there must be exactly k triples of marked squares that form an L-shaped tromino — all three squares are inside a 2×2 square.

The set of squares forms a connected figure if it is possible to get from any square to any other one if you are allowed to move from a square to any adjacent by a common side square.

Peter cannot fulfill the task, so he asks you for help. Help him to create such figure.

Input

Input data contains one or more test cases. The first line contains the number of test cases t ($1 \le t \le 100$).

Each of the following t test cases is described by a line that contains three integers: n, m and k ($3 \le n, m, n \times m \le 10^5$, $0 \le k \le 10^9$).

The sum of values of $n \times m$ for all tests in one input data doesn't exceed 10^5.

Output

For each test case print the answer.

If it is possible to create such figure, print n lines, m characters each, use asterisk '*' to denote the marked square, and dot '.' to denote the unmarked one.

If there is no solution, print `-1`.

Print empty line between test cases.

Example

input

```
3
3 3 4
3 3 5
3 3 3
```

output

```
.*.
***
.*.

**.
**.
*..

.*.
***
*..
```

D. Slalom

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

Little girl Masha likes winter sports, today she's planning to take part in slalom skiing.

The track is represented as a grid composed of $n \times m$ squares. There are rectangular obstacles at the track, composed of grid squares. Masha must get from the square $(1, 1)$ to the square (n, m). She can move from a square to adjacent square: either to the right, or upwards. If the square is occupied by an obstacle, it is not allowed to move to that square.

One can see that each obstacle can actually be passed in two ways: either it is to the right of Masha's path, or to the left. Masha likes to try all ways to do things, so she would like to know how many ways are there to pass the track. Two ways are considered different if there is an obstacle such that it is to the right of the path in one way, and to the left of the path in the other way.

Help Masha to find the number of ways to pass the track. The number of ways can be quite big, so Masha would like to know it modulo $10^9 + 7$.

The pictures below show different ways to pass the track in sample tests.

Input

The first line of input data contains three positive integers: n, m and k ($3 \le n, m \le 10^6$, $0 \le k \le 10^5$) — the size of the track and the number of obstacles.

The following k lines contain four positive integers each: x_1, y_1, x_2, y_2 ($1 \le x_1 \le x_2 \le n$, $1 \le y_1 \le y_2 \le m$) — coordinates of bottom left, and top right squares of the obstacle.

It is guaranteed that there are no obstacles at squares $(1, 1)$ and (n, m), and no obstacles overlap (but some of them may touch).

Output

Output one integer — the number of ways to pass the track modulo $10^9 + 7$.

Examples

input
```
3 3 0
```
output
```
1
```

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

input
```
5 5 3
2 2 2 3
4 2 5 2
4 4 4 4
```
output
```
3
```

E. Cipher

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

Borya has recently found a big electronic display. The computer that manages the display stores some integer number. The number has n decimal digits, the display shows the encoded version of the number, where each digit is shown using some lowercase letter of the English alphabet.

There is a legend near the display, that describes how the number is encoded. For each digit position i and each digit j the character c is known, that encodes this digit at this position. Different digits can have the same code characters.

Each second the number is increased by 1. And one second after a moment when the number reaches the value that is represented as n 9-s in decimal notation, the loud beep sounds.

Andrew knows the number that is stored in the computer. Now he wants to know how many seconds must pass until Borya can definitely tell what was the original number encoded by the display. Assume that Borya can precisely measure time, and that the encoded number will first be increased exactly one second after Borya started watching at the display.

Input

Input data contains multiple test cases. The first line of input contains t $(1 \le t \le 100)$ — the number of test cases.

Each test case is described as follows. The first line of the description contains n $(1 \le n \le 18)$ — the number of digits in the number. The second line contains n decimal digits without spaces (but possibly with leading zeroes) — the number initially stored in the display computer. The following n lines contain 10 characters each. The j-th character of the i-th of these lines is the code character for a digit j - 1 in position i, most significant digit positions are described first.

Output

For each test case print an integer: the number of seconds until Borya definitely knows what was the initial number stored on the display of the computer. Do not print leading zeroes.

Example

input

```
3
2
42
abcdefghij
jihgfedcba
2
42
aaaaaaaaaa
aaaaaaaaaa
1
2
abcdabcdff
```

output

```
0
58
2
```

F. Array Covering

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

Misha has an array of integers of length n. He wants to choose k different continuous subarrays, so that each element of the array belongs to at least one of the chosen subarrays.

Misha wants to choose the subarrays in such a way that if he calculated the sum of elements for each subarray, and then add up all these sums, the resulting value was maximum possible.

Input

The first line of input contains two integers: n, k ($1 \le n \le 100\,000$, $1 \le k \le n \cdot (n+1) / 2$) — the number of elements in the array and the number of different subarrays that must be chosen.

The second line contains n integers a_i ($-50\,000 \le a_i \le 50\,000$) — the elements of the array.

Output

Output one integer — the maximum possible value Misha can get by choosing k different subarrays.

Example

input

```
5 4
6 -4 -10 -4 7
```

output

```
11
```