A. Square Earth?

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

Meg the Rabbit decided to do something nice, specifically — to determine the shortest distance between two points on the surface of our planet. But Meg... what can you say, she wants everything simple. So, she already regards our planet as a two-dimensional circle. No, wait, it's even worse — as a square of side n. Thus, the task has been reduced to finding the shortest path between two dots on a square (the path should go through the square sides). To simplify the task let us consider the vertices of the square to lie at points whose coordinates are: $(0, 0)$, $(n, 0)$, $(0, n)$ and (n, n).

Input

The single line contains 5 space-separated integers: n, x_1, y_1, x_2, y_2 $(1 \le n \le 1000, 0 \le x_1, y_1, x_2, y_2 \le n)$ which correspondingly represent a side of the square, the coordinates of the first point and the coordinates of the second point. It is guaranteed that the points lie on the sides of the square.

Output

You must print on a single line the shortest distance between the points.

Sample test(s)

input

```
2 0 0 1 0
```

output

```
1
```

input

```
2 0 1 2 1
```

output

```
4
```

input

```
100 0 0 100 100
```

output

```
200
```

B. Martian Architecture

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

Chris the Rabbit found the traces of an ancient Martian civilization. The brave astronomer managed to see through a small telescope an architecture masterpiece — "A Road to the Sun". The building stands on cubical stones of the same size. The foundation divides the entire "road" into cells, into which the cubical stones are fit tightly. Thus, to any cell of the foundation a coordinate can be assigned. To become the leader of the tribe, a Martian should build a Road to the Sun, that is to build from those cubical stones on a given foundation a stairway. The stairway should be described by the number of stones in the initial coordinate and the coordinates of the stairway's beginning and end. Each following cell in the coordinate's increasing order should contain one cubical stone more than the previous one. At that if the cell has already got stones, they do not count in this building process, the stairways were simply built on them. In other words, let us assume that a stairway is built with the initial coordinate of l, the final coordinate of r and the number of stones in the initial coordinate x. That means that x stones **will be added** in the cell l, $x + 1$ stones will be added in the cell $l + 1$, ..., $x + r - l$ stones will be added in the cell r.

Chris managed to find an ancient manuscript, containing the descriptions of all the stairways. Now he wants to compare the data to be sure that he has really found "A Road to the Sun". For that he chose some road cells and counted the total number of cubical stones that has been accumulated throughout the Martian history and then asked you to count using the manuscript to what the sum should ideally total.

Input

The first line contains three space-separated integers: n, m, k ($1 \le n, m \le 10^5$, $1 \le k \le min(n, 100)$) which is the number of cells, the number of "Roads to the Sun" and the number of cells in the query correspondingly. Each of the following m roads contain three space-separated integers: a_i, b_i, c_i ($1 \le a_i \le b_i \le n$, $1 \le c_i \le 1000$) which are the stairway's description, its beginning, end and the initial cell's height. Then follow a line, containing k different space-separated integers b_i. All these numbers ranging from 1 to n are cells, the number of stones in which interests Chris.

Output

You have to print a single number on a single line which is the sum of stones in all the cells Chris is interested in.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cin` (also you may use `%I64d`).

Sample test(s)

input

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

output

```
5
```

input

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

output

```
4
```

input

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

output

```
6
```

C. Array

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

Chris the Rabbit has been interested in arrays ever since he was a child. At the moment he is researching arrays with the length of n, containing only integers from 1 to n. He is not good at math, that's why some simple things drive him crazy. For example, yesterday he grew keen on counting how many different beautiful arrays there are. Chris thinks that an array is beautiful if it meets one of the two conditions:

- each elements, starting from the second one, is no more than the preceding one
- each element, starting from the second one, is no less than the preceding one

Having got absolutely mad at himself and at math, Chris came to Stewie and Brian to ask them for help. However, they only laughed at him and said that the answer is too simple and not interesting. Help Chris the Rabbit to find the answer at last.

Input

The single line contains an integer n which is the size of the array ($1 \leq n \leq 10^5$).

Output

You must print the answer on a single line. As it can be rather long, you should print it modulo 1000000007.

Sample test(s)

input

```
2
```

output

```
4
```

input

```
3
```

output

```
17
```

D. Journey

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

Stewie the Rabbit explores a new parallel universe. This two dimensional universe has the shape of a rectangular grid, containing n lines and m columns. The universe is very small: one cell of the grid can only contain one particle. Each particle in this universe is either static or dynamic. Each static particle always remains in one and the same position. Due to unintelligible gravitation laws no two static particles in the parallel universe can be present in one column or row, and they also can't be present in the diagonally **adjacent** cells. A dynamic particle appears in a random empty cell, randomly chooses the destination cell (destination cell may coincide with the start cell, see the samples) and moves there along the shortest path through the cells, unoccupied by the static particles. All empty cells have the same probability of being selected as the beginning or end of the path. Having reached the destination cell, the particle disappears. Only one dynamic particle can exist at one moment of time. This particle can move from a cell to a cell if they have an adjacent side, and this transition takes exactly one galactic second. Stewie got interested in what is the average lifespan of one particle in the given universe.

Input

The first line contains two space-separated integers: n, m ($2 \leq n, m \leq 1000$) which represent the sizes of the universe. The next n lines containing m symbols each describe the universe without dynamic particles — the j-th symbol of the i-th line equals to `'X'` if the cell is occupied by a static particle, and to `'.'` if it is empty. It is guaranteed that the described universe satisfies the properties described above, that is no two static particles can be in one column or in one row, besides, they can't be positioned in the diagonally adjacent cells.

Output

You have to print on a single line a single number which is the average life span of a particle with an accuracy of at least 6 decimal places.

The answer will be accepted if it is within 10^{-6} of absolute or relative error from the correct answer.

Sample test(s)

input

```
2 2
..
.X
```

output

```
0.888888888889
```

input

```
3 3
...
.X.
...
```

output

```
2.000000000000
```

E. Chess

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

Brian the Rabbit adores chess. Not long ago he argued with Stewie the Rabbit that a knight is better than a king. To prove his point he tries to show that the knight is very fast but Stewie doesn't accept statements without evidence. He constructed an infinite chessboard for Brian, where he deleted several squares to add some more interest to the game. Brian only needs to count how many different board squares a knight standing on a square with coordinates of $(0, 0)$ can reach in no more than k moves. Naturally, it is forbidden to move to the deleted squares.

Brian doesn't very much like exact sciences himself and is not acquainted with programming, that's why he will hardly be able to get ahead of Stewie who has already started solving the problem. Help Brian to solve the problem faster than Stewie.

Input

The first line contains two integers k and n ($0 \le k \le 10^{18}, 0 \le n \le 440$) which are correspondingly the maximal number of moves a knight can make and the number of deleted cells. Then follow n lines, each giving the coordinates of a deleted square in the form (x_i, y_i) ($|x_i| \le 10, |y_i| \le 10$). All the numbers are integer, the deleted squares are different and it is guaranteed that the square $(0, 0)$ is not deleted.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cin` (also you may use `%I64d`).

Output

You must print the answer on a single line. As it can be rather long, you should print it modulo 1000000007.

Sample test(s)

input

```
1 0
```

output

```
9
```

input

```
2 7
-1 2
1 2
2 1
2 -1
1 -2
-1 -2
-2 -1
```

output

```
9
```