A. Counting Kangaroos is Fun

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

There are n kangaroos with pockets. Each kangaroo has a size (integer number). A kangaroo can go into another kangaroo's pocket if and only if the size of kangaroo who hold the kangaroo is at least twice as large as the size of kangaroo who is held.

Each kangaroo can hold at most one kangaroo, and the kangaroo who is held by another kangaroo cannot hold any kangaroos.

The kangaroo who is held by another kangaroo cannot be visible from outside. Please, find a plan of holding kangaroos with the minimal number of kangaroos who is visible.

Input

The first line contains a single integer — n $(1 \le n \le 5 \cdot 10^5)$. Each of the next n lines contains an integer s_i — the size of the i-th kangaroo $(1 \le s_i \le 10^5)$.

Output

Output a single integer — the optimal number of visible kangaroos.

Sample test(s)

input

```
8
2
5
7
6
9
8
4
2
```

output

```
5
```

input

```
8
9
1
6
2
6
5
8
3
```

output

```
5
```

B. Counting Rectangles is Fun

time limit per test: 4 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

There is an $n \times m$ rectangular grid, each cell of the grid contains a single integer: zero or one. Let's call the cell on the i-th row and the j-th column as (i, j).

Let's define a "rectangle" as four integers a, b, c, d $(1 \le a \le c \le n;\ 1 \le b \le d \le m)$. Rectangle denotes a set of cells of the grid $\{(x, y):\ a \le x \le c, b \le y \le d\}$. Let's define a "good rectangle" as a rectangle that includes only the cells with zeros.

You should answer the following q queries: calculate the number of good rectangles all of which cells are in the given rectangle.

Input

There are three integers in the first line: n, m and q $(1 \le n, m \le 40,\ 1 \le q \le 3 \cdot 10^5)$. Each of the next n lines contains m characters — the grid. Consider grid rows are numbered from top to bottom, and grid columns are numbered from left to right. Both columns and rows are numbered starting from 1.

Each of the next q lines contains a query — four integers that describe the current rectangle, a, b, c, d $(1 \le a \le c \le n;\ 1 \le b \le d \le m)$.

Output

For each query output an answer — a single integer in a separate line.

Sample test(s)

input

```
5 5 5
00101
00000
00001
01000
00001
1 2 2 4
4 5 4 5
1 2 5 2
2 2 4 5
4 2 5 3
```

output

```
10
1
7
34
5
```

input

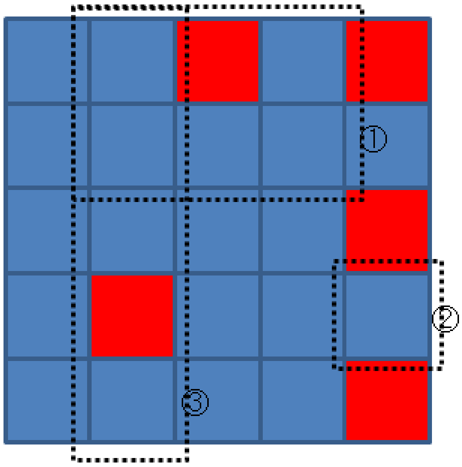
```
4 7 5
0000100
0000010
0011000
0000000
1 7 2 7
3 1 3 1
2 3 4 5
1 2 2 7
2 2 4 7
```

output

```
3
1
16
27
52
```

Note

For the first example, there is a 5×5 rectangular grid, and the first, the second, and the third queries are represented in the following image.



- For the first query, there are 10 good rectangles, five 1×1, two 2×1, two 1×2, and one 1×3.
- For the second query, there is only one 1×1 good rectangle.
- For the third query, there are 7 good rectangles, four 1×1, two 2×1, and one 3×1.

C. Watching Fireworks is Fun

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A festival will be held in a town's main street. There are n sections in the main street. The sections are numbered 1 through n from left to right. The distance between each adjacent sections is 1.

In the festival m fireworks will be launched. The i-th ($1 \le i \le m$) launching is on time t_i at section a_i. If you are at section x ($1 \le x \le n$) at the time of i-th launching, you'll gain happiness value $b_i - |a_i - x|$ (note that the happiness value might be a negative value).

You can move up to d length units in a unit time interval, but it's prohibited to go out of the main street. Also you can be in an arbitrary section at initial time moment (time equals to 1), and want to maximize the sum of happiness that can be gained from watching fireworks. Find the maximum total happiness.

Note that two or more fireworks can be launched at the same time.

Input

The first line contains three integers n, m, d ($1 \le n \le 150000$; $1 \le m \le 300$; $1 \le d \le n$).

Each of the next m lines contains integers a_i, b_i, t_i ($1 \le a_i \le n$; $1 \le b_i \le 10^9$; $1 \le t_i \le 10^9$). The i-th line contains description of the i-th launching.

It is guaranteed that the condition $t_i \le t_{i+1}$ ($1 \le i < m$) will be satisfied.

Output

Print a single integer — the maximum sum of happiness that you can gain from watching all the fireworks.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Sample test(s)

input

```
50 3 1
49 1 1
26 1 4
6 1 10
```

output

```
-31
```

input

```
10 2 1
1 1000 4
9 1000 4
```

output

```
1992
```

D. Choosing Subtree is Fun

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

There is a tree consisting of n vertices. The vertices are numbered from 1 to n.

Let's define the length of an interval $[l, r]$ as the value $r - l + 1$. The score of a subtree of this tree is the maximum length of such an interval $[l, r]$ that, the vertices with numbers $l, l + 1, ..., r$ belong to the subtree.

Considering all subtrees of the tree whose size is at most k, return the maximum score of the subtree. Note, that in this problem tree is not rooted, so a subtree — is an arbitrary connected subgraph of the tree.

Input

There are two integers in the first line, n and k ($1 \le k \le n \le 10^5$). Each of the next $n - 1$ lines contains integers a_i and b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$). That means a_i and b_i are connected by a tree edge.

It is guaranteed that the input represents a tree.

Output

Output should contain a single integer — the maximum possible score.

Sample test(s)

input

```
10 6
4 10
10 6
2 9
9 6
8 5
7 1
4 7
7 3
1 8
```

output

```
3
```

input

```
16 7
13 11
12 11
2 14
8 6
9 15
16 11
5 14
6 15
4 3
11 15
15 14
10 1
3 14
14 7
1 7
```

output

```
6
```

Note

For the first case, there is some subtree whose size is at most 6, including 3 consecutive numbers of vertices. For example, the subtree that consists of $\{1, 3, 4, 5, 7, 8\}$ or of $\{1, 4, 6, 7, 8, 10\}$ includes 3 consecutive numbers of vertices. But there is no subtree whose size is at most 6, which includes 4 or more consecutive numbers of vertices.

E. Drawing Circles is Fun

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

There are a set of points S on the plane. This set doesn't contain the origin $O(0, 0)$, and for each two distinct points in the set A and B, the triangle OAB has strictly positive area.

Consider a set of pairs of points $(P_1, P_2), (P_3, P_4), ..., (P_{2k-1}, P_{2k})$. We'll call the set *good* if and only if:

- $k \geq 2$.
- All P_i are distinct, and each P_i is an element of S.
- For any two pairs (P_{2i-1}, P_{2i}) and (P_{2j-1}, P_{2j}), the circumcircles of triangles $OP_{2i-1}P_{2j-1}$ and $OP_{2i}P_{2j}$ have a single common point, and the circumcircle of triangles $OP_{2i-1}P_{2j}$ and $OP_{2i}P_{2j-1}$ have a single common point.

Calculate the number of good sets of pairs modulo $1000000007\ (10^9 + 7)$.

Input

The first line contains a single integer n $(1 \leq n \leq 1000)$ — the number of points in S. Each of the next n lines contains four integers a_i, b_i, c_i, d_i $(0 \leq |a_i|, |c_i| \leq 50;\ 1 \leq b_i, d_i \leq 50;\ (a_i, c_i) \neq (0, 0))$. These integers represent a point $\left(\frac{a_i}{b_i}, \frac{c_i}{d_i}\right)$.

No two points coincide.

Output

Print a single integer — the answer to the problem modulo $1000000007\ (10^9 + 7)$.

Sample test(s)

input

```
10
-46 46 0 36
0 20 -24 48
-50 50 -49 49
-20 50 8 40
-15 30 14 28
4 10 -4 5
6 15 8 10
-20 50 -3 15
4 34 -16 34
16 34 2 17
```

output

```
2
```

input

```
10
30 30 -26 26
0 15 -36 36
-28 28 -34 34
10 10 0 4
-8 20 40 50
9 45 12 30
6 15 7 35
36 45 -8 20
-16 34 -4 34
4 34 8 17
```

output

```
4
```

input

```
10
0 20 38 38
-30 30 -13 13
-11 11 16 16
30 30 0 37
6 30 -4 10
6 15 12 15
-4 5 -10 25
-16 20 4 10
8 17 -2 17
16 34 2 17
```

output

```
10
```