A. Champion

Time Limit: 1 second

Points: 100

Danilo is a historian, specialising in the ancient sport of Dinoderby. He has gathered records of n notable players, and estimated the ability of each player with a rating between 1 and n inclusive. Note that the ratings of two players may be equal.

Recently, Danilo has discovered that each player was active for some contiguous period between year 1 and year m. He now wonders about a hypothetical competition each year, in which the highest rated active player would be the champion, with ties broken arbitrarily. If there were no active players in a particular year, there would of course be no champion.

Help Danilo determine the rating of the champion in each year.

Input

The first line of input consists of two space-separated integers, n and m, representing the number of players and the number of years respectively. n lines follow, the ith consisting of three space-separated integers r_i, a_i and b_i, representing player i's rating, start year and end year respectively.

Constraints

All input will satisfy the following constraints:

- $1 \leq n, m, \leq 100,000$.
- For all $1 \leq i \leq n$, we have $1 \leq r_i \leq n$ and $1 \leq a_i \leq b_i \leq m$.

Output

Output m lines, the ith consisting of a single integer, representing the rating of the champion in year i. If no players were active in year i, output 0 instead.

Subtasks

A1 (50 points): $1 \leq n, m \leq 5,000$.

A2 (50 points): no restrictions.

Sample Input 1

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

Sample Output 1

```
2
2
1
2
2
```

Sample Input 2

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

Sample Output 2

```
2
2
1
0
```

Explanation

In sample case 1, there are three players to consider over five years. In years 1, 2, 4 and 5, the highest rated active player has a rating of 2. However, in year 3, only player 3 is active, so the correct output is 1.

In sample case 2, there are two players to consider over four years. Player 1 is the champion for the first two years, after which they become inactive. Player 3 is the champion in the third year. In the fourth year, no players are active, so there is no champion.

B. Partners

Time Limit: 1 second

Points: 100

Curt and Owen are a pair of spies who have infiltrated a secret facility. They want to walk across a network of beams spanning the ceiling. This network consists of n junctions, with m beams each joining a pair of junctions. You are guaranteed that no two beams intersect, except instances where both beams have one endpoint in common.

Both spies begin at junction 1 with coordinates (x_1, y_1), and they must both make it to junction n at (x_n, y_n).

They suspect that the beams are weak, so they will only stand still at junctions. The junctions are also not very sturdy, so they will not stand at the same junction simultaneously (except at junctions 1 and n).

However, they must be careful not to be spotted! While either spy walks a beam, his partner will stand still at a junction and keep watch. Their radios have a fixed range, so they agree to keep the straight line distance between them at most d.

What is the minimum total distance that both spies can walk?

Input

The first line of input consists of three space-separated integers n, m and d, representing the number of junctions, number of beams and radio range respectively. n lines follow, the ith of which consists of two space-separated integers x_i and y_i, representing the coordinates of the ith junction. m lines follow, the jth of which consists of two space-separated integers a_j and b_j, representing the junctions joined by the jth beam.

Constraints

All input will satisfy the following constraints:

- $2 \leq n \leq 200$.
- $n - 1 \leq m \leq \max(3n - 6, n - 1)$.
- $1 \leq d \leq 2 \times 10^9$.
- For all $1 \leq i \leq n$, $0 \leq x_i, y_i < 10^9$.
- For all $1 \leq j \leq m$, $1 \leq a_j < b_j \leq n$.

Output

Output a single real number, the minimum total distance walked by both spies, or -1 if it is impossible for both spies to reach junction n.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}. Formally, if your program outputs a number a when the official answer is b, you answer will be accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Subtasks

B1 (50 points): $d = 2 \times 10^9$.

B2 (50 points): no restrictions.

Sample Input 1

```
6 7 2000000000
0 0
2 0
0 2
5 3
3 5
5 5
1 2
1 3
2 4
2 5
3 5
4 6
5 6
```

Sample Output 1

```
16.485281374
```

Sample Input 2

```
5 7 2000000000
0 0
0 6
3 3
6 6
6 0
1 2
1 3
2 3
2 4
3 4
3 5
4 5
```

Sample Output 2

```
16.970562748
```

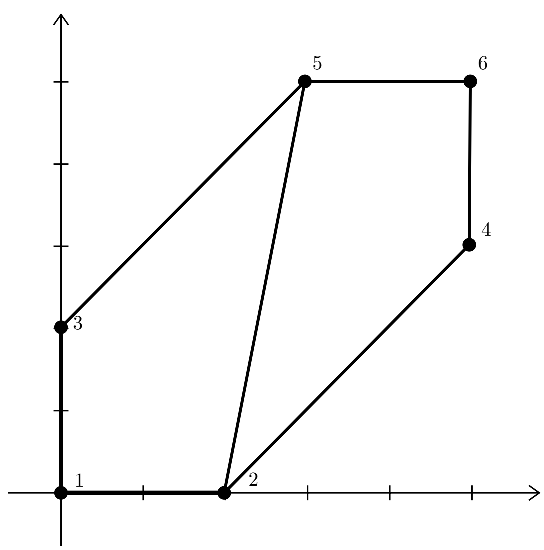
Sample Input 3

```
5 7 5
0 0
0 6
3 3
6 6
6 0
1 2
1 3
2 3
2 4
3 4
3 5
4 5
```

Sample Output 3

```
26.485281374
```

Explanation

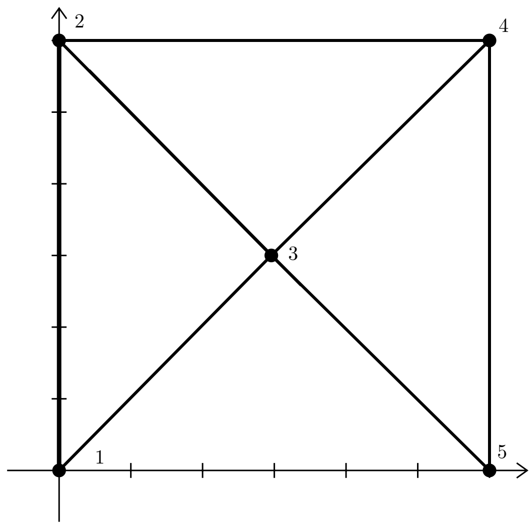


In sample case 1, there are 6 junctions and 7 beams. Both spies can go from junction 1 to junction 6 by a sequence of moves such as:

1. Curt walks beam 1, from junction 1 to junction 2

2. Owen walks beam 2, from junction 1 to junction 3
3. Curt walks beam 3, from junction 2 to junction 4
4. Owen walks beam 5, from junction 3 to junction 5
5. Curt walks beam 6, from junction 4 to junction 6
6. Owen walks beam 7, from junction 5 to junction 6.

The total distance walked is $8 + 6\sqrt{2}$, which is the minimum possible distance.



In sample case 2, there are 5 junctions and 7 beams. Junctions 1, 2, 4 and 5 form a square of side length 6 with junction 3 in the centre of the square. Since the radio range is very large, they can use the following sequence of moves:

1. Curt walks beam 2, from junction 1 to junction 3
2. Curt walks beam 6, from junction 3 to junction 5
3. Owen walks beam 2, from junction 1 to junction 3
4. Owen walks beam 6. from junction 3 to junction 5.

Note that Curt had to leave junction 3 before Owen could arrive there - they cannot both be at junction 3 simultaneously. The total distance walked is $12\sqrt{2}$, which is the minimum possible distance.

The only change from sample case 2 to sample case 3 is the radio range, which is now restricted to 5 units. Suppose Curt walks first; he cannot go to junction 2 as it would put him out of range from Owen. He instead goes to junction 3, following beam 2. Curt cannot continue on using beam 6 as in sample case 2, because again he would be out of range from Owen. Instead, Owen can move:

1. along beam 1, from junction 1 to junction 2, then
2. along beam 4, from junction 2 to junction 4, then
3. along beam 7, from junction 4 to junction 5.

Finally, Curt can now walk beam 6 to join Owen at junction 5. Curt walked

$6\sqrt{2}$ units, and Owen walked 18 uniits. The total of $18 + 6\sqrt{2}$ is the minimum possible distance.

Note that sample case 3 is not a valid input for subtask B1.

C. Bot Factory

Time Limit: 1 second

Points: 100

Ágata is in charge of managing a factory making uber-bots. The factory is represented as a series of n work-stations connected by m one way conveyor belts, each of which take exactly one second to traverse. The goal is to maximise the factory's output of level k bots.

A station receives bots on its incoming conveyor belts, and automatically routes each bot to its outgoing conveyor belts. Furthermore, some stations are marked as 'source', 'sink', or 'upgrader' stations.

- Source stations generate one level 1 bot every second.
- If a level k bot arrives at at a sink station, it will be marked as complete and removed from the factory.
- Each upgrade station upgrades only bots of a specified level. If a level q bot arrives at a q-upgrader station, it is automatically upgraded to level $q + 1$. Note that bots of levels other than q are not upgraded at a q-upgrader station.

There is one catch. While multiple bots can travel along the same conveyor belt at the same time, no two of those bots can be of the same level. Note that this restriction applies only to the conveyor belts; each station can process multiple bots of the same level at the same time.

Ágata has tasked you with programming how each station routes every bot. Once production has started, you can no longer change this routing. Furthermore, you are allowed to turn off some of the source stations so that they do not produce bots. What is the maximum number of bots per second that the factory can output?

Input

The first line of input consists of three space-separated integers, n, m and k, representing the number of stations, the number of conveyor belts and the level at which bots are accepted respectively. The second line of input consists of three space-separated integers, x, y and z, representing the number of source stations, sink stations and upgrader stations respectively. The third line consists of x space-separated integers $s_1, s_2, \ldots, s_x$, representing the source stations. The fourth line consists of y space-separated integers $t_1, t_2, \ldots, t_y$, representing the sink stations. z lines follow, the ith consisting of two space-separated integers p_i and q_i, representing a q_i-upgrader station p_i. m lines follow, the jth consisting of two space-separated integers a_j and b_j, representing a conveyor belt from station a_j to station b_j.

Constraints

All input will satisfy the following constraints:

- $2 \le n \le 600$.
- $1 \le m \le 600$.
- $1 \le k \le 150$.
- $x, y \ge 1$.
- $z \ge 0$.
- $x + y + z \le n$.
- The s_i, t_i and p_i are distinct, and all between 1 and n inclusive.
- For all $1 \le i \le z$, we have $1 \le q_i < k$.
- For each $1 \le j \le m$, we have $1 \le a_j, b_j \le n$ and $a_j \ne b_j$.
- All pairs (a_j, b_j) are distinct.

Output

Output one integer, the maximum number of accepted bots per second that the factory can produce.

Subtasks

C1 (50 points): $x = 1$.

C2 (50 points): no restrictions.

Sample Input 1

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

Sample Output 1

```
1
```

Sample Input 2

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

Sample Output 2

```
1
```

Sample Input 3

```
3 2 1
2 1 0
1 2
3
1 3
2 3
```

Sample Output 3

```
2
```

Sample Input 4

```
3 2 1
1 2 0
1
2 3
1 2
1 3
```

Sample Output 4

```
1
```

Explanation

The sample inputs are shown in the images below:

- Incoming red arrows correspond to sources.
- Outgoing red arrows correspond to sinks.
- Blue arrows next to stations correspond to upgrader stations, with the number being the level of the upgrader station.

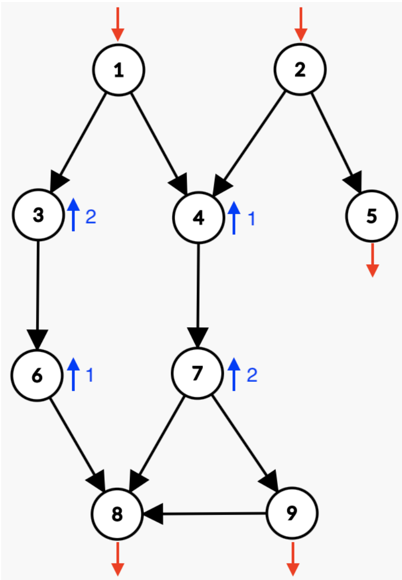


Figure 1: Sample Input 1

In Sample Input 1, either source 1 or source 2 (but not both) can produce bots. The bot can then travel to station 4, where it is upgraded to level 2, then station 7, where it is upgraded to level 3, and then finally to either station 8 or 9 (which are both sinks).

In Sample Input 2, station 1 can produce a bot which then travels to station 4 (via stations 2 and 3), where it is upgraded to level 2. The bot then travels to station 2, where it is upgraded to level 3, and it can then travel to the sink at station 5 (via stations 3 and 4).

In Sample Input 3, both stations 1 and 2 can produce bots which travel to the sink at station 3.

In Sample Input 4, only station 1 can produce a bot which travels to a sink at either station 2 or station 3.

Note that sample cases 1 and 3 are not valid inputs for subtask C1.

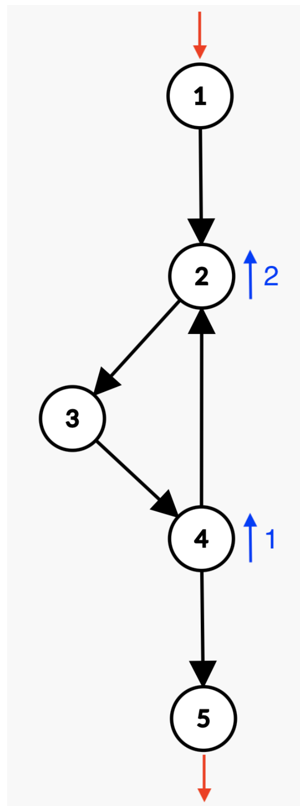


Figure 2: Sample Input 2

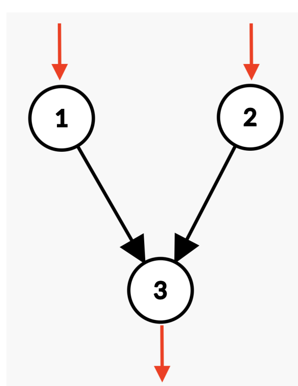


Figure 3: Sample Input 3

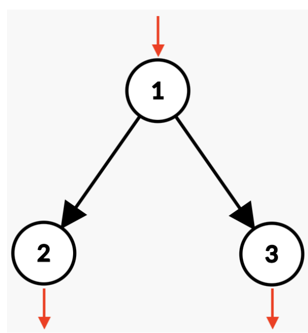


Figure 4: Sample Input 4