Task: GOD
Godzilla

Maratona de Programação, Day 4. Available memory: 256 MB. *30.01.2015*

The favourite activity of citizens of Byteburg is watching a popular soap opera *The Cold and the Pitiful*. This TV series is transmitted via cable network to each house in Byteburg. The TV cable network consists of n nodes and m **unidirectional** connections. For every node in the network, there is at least one house connected to it. The TV series is transmitted to some of the nodes (called the *transmission* nodes). You can watch the TV program in a given house if there exists a connection (not necessarily direct) from a transmission node to the node to which the house is connected. To minimize the costs, the number of transmission nodes should be as small as possible.

Unfortunately, one day a nasty monster Godzilla has come to Byteburg. Strangely enough, the monster enjoys eating TV cable infrastructure. Each day it eats one connection of the network. The owner of the TV network does not want the number of subscribers to decrease, so he has to update the set of transmission nodes in such a way that every citizen can still watch *The Cold and the Pitiful* every day. Help him check if he does it optimally.

Input

The first line of the input contains two integers n and m ($1 \leq n, m \leq 100\,000$), which specify the number of nodes and the number of connections in the cable network. The following m lines contain descriptions of the connections. Each of the lines contains two integers a and b ($1 \leq a, b \leq n, a \neq b$), which specify a unidirectional connection from the node a to the node b. There exists at most one direct connection between any pair of nodes in a given direction.

The next line contains an integer k ($1 \leq k \leq m$) representing the number of connections that were attacked by Godzilla. The following k lines contain the numbers of the eaten connections (the connections are numbered from 1 in the same order as they are listed in the input). Every connection appears in the list at most once.

Output

Your program should output exactly k lines. The i-th of these lines should contain a single integer, representing the number of transmission nodes to which the TV series should be transmitted after the i-th Godzilla's attack.

Example

For the input data:

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

the correct result is:

```
2
3
```

Task: KOZ
The Goat

Maratona de Programação, Day 4. Available memory: 256 MB. *30.01.2015*

Farmer Byteman took his sledgehammer and drove n stakes into the ground of an infinite pasture. Through the next k days each morning farmer Byteman takes his goat out to the pasture and ties it to a randomly selected stake with a cord of length l. During the day the goat eats all the grass in its reach. Unfortunately for the goat, the grass does not grow back. Moreover, it can happen that the absentminded farmer will tie the goat more than once to the same stake.

What is the expected value of the area of the pasture on which the grass will be eaten after k days?

Input

The first line of the input contains three integers n, k and l ($1 \leq n, k, l \leq 1\,000$) specifying respectively the number of stakes, the number of days, and the length of the cord. Each of the next n lines contains the coordinates of one stake in the form of a pair of integers x_i, y_i ($-1\,000 \leq x_i, y_i \leq 1\,000$). No two stakes are driven into the ground in the same place.

Output

In the only line of the output a single real number should be written. The number should be the expected value of the area of the part of the pasture from which the goat will have eaten the grass during k days. The answer will be accepted if it will differ from the correct answer by no more than 10^{-6}. No more than 20 digits should be given after the decimal point.

Example

For the input data:

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

the correct result is:

```
4.098204131080311
```

Explanation of the example: If the goat will be tied to the same stake during both days, the area of the eaten grass will be equal to π. On the other hand if it will be tied to two different stakes, the area will be equal to $\frac{4}{3}\pi + \frac{\sqrt{3}}{2}$. Thus the answer is $\frac{7}{6}\pi + \frac{\sqrt{3}}{4}$.

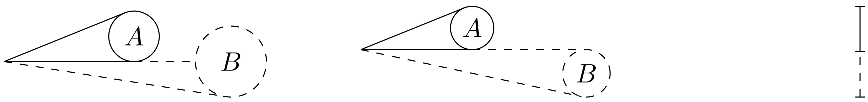
Task: PLA
Planets

Maratona de Programação, Day 4. Available memory: 256 MB. *30.01.2015*

Imagine that you are living on a plane in point $(0, 0)$. You observe on the sky two planets A and B. Planet A is a circle of origin (x_1, y_1) and radius r_1, and planet B is a circle of origin (x_2, y_2) and radius r_2. Planet A is always fully visible, but planet B can be obscured by planet A. There are four types of this situation:

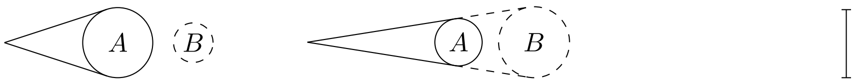
1. Planet B is fully visible. (On the left the situation is depicted, on the right is shown how we see the planets.)

2. Planet B is obscured by planet A on one side.



3. Planet A obscures the middle of planet B, i.e. we see both boundaries of planet B, but we do not see it in full.

4. Planet A fully obscures planet B, i.e. we cannot see planet B.



Your task is to write a program, which calculates what type of situations we are dealing with.

Input

In the only line of the input there are six integers x_1, y_1, r_1, x_2, y_2 and r_2 $(1 \leq x_1, y_1, r_1, x_2, y_2, r_2 \leq 1000)$, specifying the positions and sizes of the planets.

You can assume that the planets are disjoint, i.e. the distance between their origins is greater that $r_1 + r_2$; planets do not intersect with point $(0, 0)$; planet A is fully visible, i.e. no point of planet A is obscured by planet B.

Output

The only line of the output should consists of one integer from 1 to 4, standing for the type of situation.

Example

For the input data:
```
1 4 1 5 7 1
```
the correct result is:
```
1
```

For the input data:
```
2 3 1 6 7 2
```
the correct result is:
```
2
```

For the input data:
```
4 3 1 10 7 4
```
the correct result is:
```
3
```

For the input data:
```
2 2 1 5 6 2
```
the correct result is:
```
4
```

Task: INT
Clients

Maratona de Programação, Day 4. Available memory: 256 MB. *30.01.2015*

A group of n clients formed before clerk's window. The clients are very patient, but the clerk gets easily frustrated. He wants to serve all the clients as fast as possible. Unfortunately, the pressure causes that the longer he works, the less efficient he become. To serve the i-th client he needs $a_i t + b_i$ minutes, if he starts serving him in time t.

Determine the order of serving the clients, which minimize the total time of serving them.

Input

In the first line of the input there is an integer n specifying the number of clients ($1 \leq n \leq 10^6$). In the next n lines there are descriptions of clients; i-th of these rows contains two integers a_i and b_i ($0 \leq a_i, b_i \leq 10^9$), specifying the time to serve the i-th client.

Output

In the only line of the output you have to write a permutation of n integers from 1 to n, describing the optimal order of serving the clients. If there is more than one optimal permutation, you have to write the lexicographically smallest.

Example

For the input data:

```
2
3 4
4 5
```

the correct result is:

```
2 1
```

Task: DNA
DNA

Maratona de Programação, Day 4. Available memory: 256 MB. *30.01.2015*

The genetic material of Recurse Uncomplicated consists of one DNA string which has an even number of n purines. There are four of such purines, we will denote them by letters `A`, `C`, `G` and `T`. Professor Bajthony discovered a relationship between DNA of Recurse and DNA of its parents. In the DNA of Recurse there are exactly $n/2$ purines which came from DNA of its father and exactly $n/2$ purines which came from DNA of its mother. Moreover, these purines appear at the same locations in parent's DNA and in offspring's DNA. On the following picture purines from respective parents are underlined.

mother	`ATGGCA`
father	`CTTCAT`
offspring	`ATGCAA`

Professor observed that despite of big number of possibilities in which Recurse's DNA can arise, the offspring of a given pair of parents have always the same DNA. Moreover, the professor hypothesize that the resulting DNA string is lexicographically smallest possibility. Your task is to write a program which will help in verification of this hypothesis.

Input

In the first line of the input there is one even integer n ($2 \leq n \leq 1\,000\,000$), specifying the length of DNA string. Each of the next two lines consists of one string of length n with letters `A`, `C`, `G` and `T`. These are descriptions of DNA of pair of Recurses.

Output

In the only line of the output you should write a n-letter string denoting the lexicographically smallest DNA string which can arise from DNA of two Recurses from the input.

Example

For the input data:

```
6
ATGGCA
CTTCAT
```

the correct result is:

```
ATGCAA
```

Task: TRI
Trip

Maratona de Programação, Day 4. Available memory: 256 MB. *30.01.2015*

A network consisting of n cities and $n-1$ roads is given. Each city is labeled with a distinct integer from 1 to n. Roads connect cities in such a way that each distinct pair of cities is connected either by a direct road or through a path consisting of direct roads. There is exactly one way to reach any city from any other city. Each city also has its own *attractiveness level*, which will be denoted by an integer. City P is more attractive than city Q if the attractiveness level of city P is strictly greater than the attractiveness level of city Q.

You are planning a trip to visit some of the most attractive cities. You want to select cities to visit based on the following requirements:

- At most k cities can be included in the trip plan.
- It must be possible to travel among the cities included in the trip plan without having to travel through cities that have been excluded from the trip plan.
- None of the cities included in the trip plan can be less attractive than any of the excluded cities. City attractiveness levels do not have to be unique, though, so it is permissible to visit only a subset of cities that are equally attractive.

The goal is to maximize the number of cities selected while satisfying the above requirements.

Input

In the first line of the input there are two integers n and k ($1 \le k \le n \le 100\,000$), specifying the number of cities and the maximal number of cities in the trip plan. The i-th of the next n lines contains two integers c_i and d_i ($1 \le c_i \le n$, $0 \le d_i \le 1\,000\,000$). The number d_i is the attractiveness level of i-th city. If $c_i \neq i$, then it means that cities i and c_i are connected with a direct road.

Output

Write one integer to the output: the maximum number of cities in the plan.

Example

For the input data:

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

the correct result is:

```
2
```

Explanation of the example: We can select a maximum number of two cities: either 3 and 1 or 3 and 5. In both cases, the attractiveness levels of the selected cities are greater than or equal to 6 and the attractiveness levels of the excluded cities are less than or equal to 6. (If, however, $k = 5$, the maximum number of cities we could select according to the rules above is four: we must select cities 3, 1, 5 and 6.)

Task: MUZ
The Museum

Maratona de Programação, Day 4. Available memory: 256 MB. *30.01.2015*

Bytemon, a well known burglar, wants to rob the National Museum of Byteotia. He is particularly interested in the Royal Family Jewels, which are displayed in the most magnificent hall of the museum. There are n exhibits watched over by m guards in this hall. The museum's custodian wanted to ensure that the visitors, admiring exhibits, are not being disturbed by the guards more than necessary. Therefore, he ordered the guards to stand in one place all the time, and look in one direction only.

Bytemon managed to get plan of this hall, where all the exhibits have been marked, as well as the arrangement of the guards. He obtained a pricing on all displayed jewels from a jeweller he knew. He also learned how much it would cost to discretely persuade each guard to close his eyes to Bytemon's activities at the time of the burglary.

Bytemon is wondering now how rich he can get. Therefore, he wants to chose the guards to be bribed, in such a way that the total value of the jewellery that is not in sight of any of guards that have not been bribed, less the cost of bribing selected guards, is as large as possible.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 200\,000$), specifying the number of exhibits and the number of guards. To describe their positioning assume that the museum plan has a rectangular coordinate system imposed. The second line of the input contains two integers w and h ($1 \le w, h \le 10^9$), describing the field of vision of the guards. Each of the guards is looking in the direction of decreasing y-coordinates, and the tangent of half of the angle of his view amounts to w/h. For simplicity, we assume that the guards and the exhibits are of a negligible size. The guard is observing all the exhibits, which are located in his field of vision (including the edge), even in case they are occluded by other exhibits or guards.

Subsequent n lines describe the position of the exhibits; the i-th of these lines contains three integers x_i, y_i, v_i ($-10^9 \le x_i, y_i \le 10^9$, $1 \le v_i \le 10^9$) indicating that the i-th exhibit has a value of v_i bytecoins and is located at the point (x_i, y_i). Subsequent m lines describe the guard positions, analogically. (However, v_i denotes the amount, in bytecoins, to be paid by Bytemon to bribe the i-th guard.) At each point there can be at most one guard or exhibit.

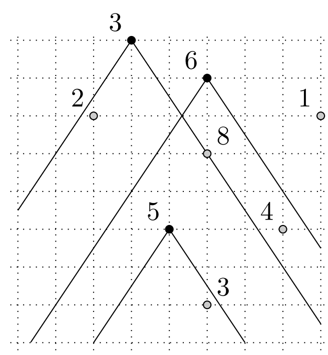
Output

Your program should output a single line containing a single integer indicating the maximum profit, in bytecoins, that could be achieved by Bytemon.

Example

For the input data:

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



the correct result is:

```
6
```

Explanation of the example: Angle of vision of each of the guards slightly exceeds 67°. Bytemon should bribe two guards, paying $3 + 6$ bytecoins, and take exhibits of a value of $2 + 8 + 4 + 1$ bytecoins.

Task: PRI
Prison Escape

Maratona de Programação, Day 4. Available memory: 256 MB. *30.01.2015*

There are $n + 1$ intersections in a prison, connected by n corridors, and one can move between any two intersections using the corridors. Intersections with only one corridor are located near the exits from the prison.

There is a cell block near every intersection. Exactly m cell blocks hold prisoners; others are empty. Due to a malfunction in the locking system, all the cells have been opened. The situation is critical and the governor of the prison needs to know where to position the guards in order to prevent the prisoners from escaping.

A guard can be located at an intersection, but not at an intersections near to a cell block which initially held prisoners. Prisoners from a cell block located near intersection X can escape from the prison if there is a path from X to an intersection located near an exit from the prison and there are no guards at any intersection along this path. The governor would like to know the minimum number of guards he must deploy in the operation.

Input

In the first line of the input there are two integers n and m ($1 \leq n \leq 200\,000$, $0 \leq m \leq n + 1$), specifying the number of corridors and the number of cell blocks holding prisoners. In the second line there is a sequence of m integers $c_1, c_2, \ldots, c_m$ ($1 \leq c_i \leq n$), specifying the numbers of intersections near which are located cells holding prisoners. The next n lines gives information about corridors; the i-th of these lines consists of two integers a_i and b_i ($1 \leq a_i, b_i \leq n + 1$), specifying that there is a corridor between intersections a_i and b_i.

Output

In the only line of the output there should be one integer specifying the minimum number of guards that can prevent all prisoners from escaping. If there is no way to prevent an escape of some prisoner, this number should be -1.

Example

For the input data:

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

the correct result is:

```
4
```

Explanation of the example: There are two cells holding prisoners (near intersections 2 and 7). Four guards can be positioned at the intersections numbered 1, 4, 8 and 9 (another solution would be to position guards at intersections 1, 3, 8 and 9). By positioning three or fewer guards, the governor is unable to prevent some prisoners from escaping.

Task: MON
Coins

Maratona de Programação, Day 4. Available memory: 256 MB. *30.01.2015*

Joe claims that he has telekinetic abilities. This statement has shocked Stan, who is a confirmed rationalist, and he immediately wanted Joe to prove it.

Joe decided to toss a coin to show what he is capable of. He says that he can do it in such a way, that there will be exactly k times more heads than tails. Stan has written down the results for all tosses and now he wants to find the longest sequence of consecutive tosses, in which heads are exactly k times more common than tails.

Input

The first line of the input contains two integers n and k ($3 \leq n \leq 1\,000\,000$, $2 \leq k \leq n-1$). The first is the number of tosses made by Joe, whereas the meaning of the second number has already been described in the task statement. The second line contains a sequence of n characters describing the outcomes of consecutive tosses. It consists of letters `O` and `R` denoting heads and tails respectively.

Output

Your program should write one integer to the first and only line of the output equal to the length of the longest sequence of consecutive tosses, in which there are exactly k times more heads than tails. If such sequence does not exist, your program should output 0.

Example

For the input data:

```
15 3
RORROOROOROOORO
```

the correct result is:

```
8
```

Explanation of the example: The series from fifth to twelfth as well as from sixth to thirteenth toss contain exactly 6 heads and 2 tails, i.e., three times more heads than tails. There does not exist a longer sequence with this property.