Codeforces Beta Round #66

A. The Elder Trolls IV: Oblivon

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

Vasya plays The Elder Trolls IV: Oblivon. Oh, those creators of computer games! What they do not come up with! Absolutely unique monsters have been added to the The Elder Trolls IV: Oblivon. One of these monsters is Unkillable Slug. Why it is "Unkillable"? Firstly, because it can be killed with cutting weapon only, so lovers of two-handed amber hammers should find suitable knife themselves. Secondly, it is necessary to make so many cutting strokes to Unkillable Slug. Extremely many. Too many!

Vasya has already promoted his character to 80-th level and in order to gain level 81 he was asked to kill Unkillable Slug. The monster has a very interesting shape. It looks like a rectangular parallelepiped with size $x \times y \times z$, consisting of undestructable cells $1 \times 1 \times 1$. At one stroke Vasya can cut the Slug along an imaginary grid, i.e. cut with a plane parallel to one of the parallelepiped side. Monster dies when amount of parts it is divided reaches some critical value.

All parts of monster do not fall after each cut, they remains exactly on its places. I. e. Vasya can cut several parts with one cut.

Vasya wants to know what the maximum number of pieces he can cut the Unkillable Slug into striking him at most k times.

Vasya's character uses absolutely thin sword with infinite length.

Input

The first line of input contains four integer numbers x, y, z, k $(1 \le x, y, z \le 10^6, 0 \le k \le 10^9)$.

Output

Output the only number — the answer for the problem.

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

Sample test(s)

input

```
2 2 2 3
```

output

```
8
```

input

```
2 2 2 1
```

output

```
2
```

Note

In the first sample Vasya make 3 pairwise perpendicular cuts. He cuts monster on two parts with the first cut, then he divides each part on two with the second cut, and finally he divides each of the 4 parts on two.

B. Need For Brake

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

Vasya plays the Need For Brake. He plays because he was presented with a new computer wheel for birthday! Now he is sure that he will win the first place in the championship in his favourite racing computer game!

n racers take part in the championship, which consists of a number of races. After each race racers are arranged from place first to n-th (no two racers share the same place) and first m places are awarded. Racer gains b_i points for i-th awarded place, which are added to total points, obtained by him for previous races. It is known that current summary score of racer i is a_i points. In the final standings of championship all the racers will be sorted in descending order of points. Racers with an equal amount of points are sorted by increasing of the name in lexicographical order.

Unfortunately, the championship has come to an end, and there is only one race left. Vasya decided to find out what the highest and lowest place he can take up as a result of the championship.

Input

The first line contains number n $(1 \le n \le 10^5)$ — number of racers. Each of the next n lines contains s_i and a_i — nick of the racer (nonempty string, which consist of no more than 20 lowercase Latin letters) and the racer's points $(0 \le a_i \le 10^6)$. Racers are given in the arbitrary order.

The next line contains the number m $(0 \le m \le n)$. Then m nonnegative integer numbers b_i follow. i-th number is equal to amount of points for the i-th awarded place $(0 \le b_i \le 10^6)$.

The last line contains Vasya's racer nick.

Output

Output two numbers — the highest and the lowest place Vasya can take up as a result of the championship.

Sample test(s)

input

```
3
teama 10
teamb 20
teamc 40
2
10 20
teama
```

output

```
2 3
```

input

```
2
teama 10
teamb 10
2
10 10
teamb
```

output

```
2 2
```

C. LionAge II

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

Vasya plays the LionAge II. He was bored of playing with a stupid computer, so he installed this popular MMORPG, to fight with his friends. Vasya came up with the name of his character — non-empty string s, consisting of a lowercase Latin letters. However, in order not to put up a front of friends, Vasya has decided to change no more than k letters of the character name so that the new name sounded as good as possible. Euphony of the line is defined as follows: for each pair of adjacent letters x and y (x immediately precedes y) the bonus $c(x, y)$ is added to the result. Your task is to determine what the greatest Euphony can be obtained by changing at most k letters in the name of the Vasya's character.

Input

The first line contains character's name s and an integer number k ($0 \le k \le 100$). The length of the nonempty string s does not exceed 100. The second line contains an integer number n ($0 \le n \le 676$) — amount of pairs of letters, giving bonus to the euphony. The next n lines contain description of these pairs «x y c», which means that sequence xy gives bonus c (x, y — lowercase Latin letters, $-1000 \le c \le 1000$). It is guaranteed that no pair x y mentioned twice in the input data.

Output

Output the only number — maximum possible euphony of the new character's name.

Sample test(s)

input

```
winner 4
4
s e 7
o s 8
l o 13
o o 8
```

output

```
36
```

input

```
abcdef 1
5
a b -10
b c 5
c d 5
d e 5
e f 5
```

output

```
20
```

Note

In the first example the most euphony name will be $looser$. It is easy to calculate that its euphony is 36.

D. FreeDiv

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

Vasya plays FreeDiv. In this game he manages a huge state, which has n cities and m two-way roads between them. Unfortunately, not from every city you can reach any other one moving along these roads. Therefore Vasya decided to divide the state into provinces so that in every province, one could reach from every city all the cities of the province, but there are no roads between provinces.

Unlike other turn-based strategies, in FreeDiv a player has the opportunity to build tunnels between cities. The tunnels are two-way roads along which one can move armies undetected by the enemy. However, no more than one tunnel can be connected to each city. As for Vasya, he wants to build a network of tunnels so that any pair of cities in his state were reachable by some path consisting of roads and a tunnels. But at that no more than k tunnels are connected to each province (otherwise, the province will be difficult to keep in case other provinces are captured by enemy armies).

Vasya discovered that maybe he will not be able to build such a network for the current condition of the state. Maybe he'll have first to build several roads between cities in different provinces to merge the provinces. Your task is to determine the minimum number of roads Vasya needs to build so that it was possible to build the required network of tunnels in the resulting state.

Input

The first line contains three integers n, m and k ($1 \le n, k \le 10^6$, $0 \le m \le 10^6$). Each of the next m lines contains two integers. They are the numbers of cities connected by a corresponding road. No road connects city to itself and there is at most one road between each pair of cities.

Output

Print a single number, the minimum number of additional roads.

Sample test(s)

input

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

output

```
0
```

input

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

output

```
0
```

input

```
4 0 2
```

output

```
1
```

Note

In the first example only one province exists, so it is not necessary to build any tunnels or roads.

In the second example two provinces exist. It is possible to merge the provinces by building a tunnel between cities 1 and 3.

In the third example at least one additional road is necessary. For example it is possible to build additional road between cities 1 and 2 and build two tunnels between cities 1 and 3, 2 and 4 after that.

E. Morrowindows

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

Vasya plays The Elder Trolls III: Morrowindows. He has a huge list of items in the inventory, however, there is no limits on the size of things. Vasya does not know the total amount of items but he is sure that are not more than x and not less than 2 items in his inventory. A new patch for the game appeared to view inventory in n different modes. Displaying in mode i is a partition of all inventory items on pages, each of which (except for maybe the last one) shows exactly a_i items. In addition, each mode shows how many pages b_i is in a complete list. Great! Perhaps this information will be enough for Vasya to find the required number. Moreover, it is very interesting, what is the fewest number of modes in which Vasya can see inventory to determine the number of items in it?

Vasya cannot use the information that was received while looking on inventory in some mode for selection of next actions. I. e. Vasya chooses some set of modes first, and then sees all the results and determines the size.

Knowing the number of a_i, x and assuming that Vasya is very smart, check whether he can uniquely determine the number of items in his inventory, and how many modes he will need to do that if he knows numbers a_i, x and he is able to know number b_i after viewing items in mode i.

Input

The first line contains two integers n and x ($0 \le n \le 10^5$, $2 \le x \le 10^9$). The second line contains integers a_i ($1 \le a_i \le 10^9$). Some numbers among all a_i may be equal.

Output

Output the fewest amount of modes required to uniquely determine amount of items in the inventory. If there is no solution output -1.

Sample test(s)

input

```
2 4
2 3
```

output

```
2
```

input

```
1 4
2
```

output

```
-1
```

Note

In the second example Vasya is not able to determine items count uniquely because 3 items, as well as 4 items, can be displayed on two pages.

F. Plane of Tanks

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

Vasya plays the Plane of Tanks. The tanks in this game keep trying to finish each other off. But your "Pedalny" is not like that... He just needs to drive in a straight line from point A to point B on the plane. Unfortunately, on the same plane are n enemy tanks. We shall regard all the tanks as points. At the initial moment of time Pedalny is at the point A. Enemy tanks would be happy to destroy it immediately, but initially their turrets are tuned in other directions. Specifically, for each tank we know the initial rotation of the turret a_i (the angle in radians relative to the OX axis in the counterclockwise direction) and the maximum speed of rotation of the turret w_i (radians per second). If at any point of time a tank turret will be aimed precisely at the tank Pedalny, then the enemy fires and it never misses. Pedalny can endure no more than k shots. Gun reloading takes very much time, so we can assume that every enemy will produce no more than one shot. Your task is to determine what minimum speed of v Pedalny must have to get to the point B. It is believed that Pedalny is able to instantly develop the speed of v, and the first k shots at him do not reduce the speed and do not change the coordinates of the tank.

Input

The first line contains 4 numbers – the coordinates of points A and B (in meters), the points do not coincide. On the second line number n is given $(1 \le n \le 10^4)$. It is the number of enemy tanks. Each of the following n lines contain the coordinates of a corresponding tank x_i, y_i and its parameters a_i and w_i $(0 \le a_i \le 2\pi,\ 0 \le w_i \le 100)$. Numbers a_i and w_i contain at most 5 digits after the decimal point. All coordinates are integers and their absolute values do not exceed 10^5. Enemy tanks can rotate a turret in the clockwise as well as in the counterclockwise direction at the angular speed of not more than w_i. It is guaranteed that each of the enemy tanks will need at least 0.1 seconds to aim at any point of the segment AB and each of the enemy tanks is posistioned no closer than 0.1 meters to line AB. On the last line is given the number k $(0 \le k \le n)$.

Output

Print a single number with absolute or relative error no more than 10^{-4} — the minimum required speed of Pedalny in meters per second.

Sample test(s)

input

```
0 0 10 0
1
5 -5 4.71238 1
0
```

output

```
4.2441
```

input

```
0 0 10 0
1
5 -5 4.71238 1
1
```

output

```
0.0000
```