Codeforces Beta Round #67 (Div. 2)

A. Life Without Zeros

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

Can you imagine our life if we removed all zeros from it? For sure we will have many problems.

In this problem we will have a simple example if we removed all zeros from our life, it's the addition operation. Let's assume you are given this equation $a + b = c$, where a and b are positive integers, and c is the sum of a and b. Now let's remove all zeros from this equation. Will the equation remain correct after removing all zeros?

For example if the equation is $101 + 102 = 203$, if we removed all zeros it will be $11 + 12 = 23$ which is still a correct equation.

But if the equation is $105 + 106 = 211$, if we removed all zeros it will be $15 + 16 = 211$ which is not a correct equation.

Input

The input will consist of two lines, the first line will contain the integer a, and the second line will contain the integer b which are in the equation as described above ($1 \le a, b \le 10^9$). There won't be any leading zeros in both. The value of c should be calculated as $c = a + b$.

Output

The output will be just one line, you should print "YES" if the equation will remain correct after removing all zeros, and print "NO" otherwise.

Sample test(s)

input
101 102

output
YES

input
105 106

output
NO

B. Facetook Priority Wall

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

Facetook is a well known social network website, and it will launch a new feature called Facetook Priority Wall. This feature will sort all posts from your friends according to the priority factor (it will be described).

This priority factor will be affected by three types of actions:

- 1. "X `posted on` Y`'s wall`" (15 points),
- 2. "X `commented on` Y`'s post`" (10 points),
- 3. "X `likes` Y`'s post`" (5 points).

X and Y will be two distinct names. And each action will increase the priority factor between X and Y (and vice versa) by the above value of points (the priority factor between X and Y is the same as the priority factor between Y and X).

You will be given n actions with the above format (without the action number and the number of points), and you have to print all the distinct names in these actions sorted according to the priority factor with you.

Input

The first line contains your name. The second line contains an integer n, which is the number of actions ($1 \le n \le 100$). Then n lines follow, it is guaranteed that each one contains exactly 1 action in the format given above. There is exactly one space between each two words in a line, and there are no extra spaces. All the letters are lowercase. All names in the input will consist of at least 1 letter and at most 10 small Latin letters.

Output

Print m lines, where m is the number of distinct names in the input (excluding yourself). Each line should contain just 1 name. The names should be sorted according to the priority factor with you in the descending order (the highest priority factor should come first). If two or more names have the same priority factor, print them in the alphabetical (lexicographical) order.

Note, that you should output all the names that are present in the input data (excluding yourself), even if that person has a zero priority factor.

The lexicographical comparison is performed by the standard "<" operator in modern programming languages. The line a is lexicographically smaller than the line b, if either a is the prefix of b, or if exists such an i ($1 \le i \le min(|a|, |b|)$), that $a_i < b_i$, and for any j ($1 \le j < i$) $a_j = b_j$, where $|a|$ and $|b|$ stand for the lengths of strings a and b correspondently.

Sample test(s)

input

```
ahmed
3
ahmed posted on fatma's wall
fatma commented on ahmed's post
mona likes ahmed's post
```

output

```
fatma
mona
```

input

```
aba
1
likes likes posted's post
```

output

```
likes
posted
```

C. Modified GCD

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

Well, here is another math class task. In mathematics, GCD is the greatest common divisor, and it's an easy task to calculate the GCD between two positive integers.

A common divisor for two positive numbers is a number which both numbers are divisible by.

But your teacher wants to give you a harder task, in this task you have to find the greatest common divisor d between two integers a and b that is in a given range from low to $high$ (inclusive), i.e. $low \le d \le high$. It is possible that there is no common divisor in the given range.

You will be given the two integers a and b, then n queries. Each query is a range from low to $high$ and you have to answer each query.

Input

The first line contains two integers a and b, the two integers as described above ($1 \le a, b \le 10^9$). The second line contains one integer n, the number of queries ($1 \le n \le 10^4$). Then n lines follow, each line contains one query consisting of two integers, low and $high$ ($1 \le low \le high \le 10^9$).

Output

Print n lines. The i-th of them should contain the result of the i-th query in the input. If there is no common divisor in the given range for any query, you should print `-1` as a result for this query.

Sample test(s)

input
9 27 3 1 5 10 11 9 11
output
3 -1 9

D. Big Maximum Sum

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

Ahmed and Mostafa used to compete together in many programming contests for several years. Their coach Fegla asked them to solve one challenging problem, of course Ahmed was able to solve it but Mostafa couldn't.

This problem is similar to a standard problem but it has a different format and constraints.

In the standard problem you are given an array of integers, and you have to find one or more consecutive elements in this array where their sum is the maximum possible sum.

But in this problem you are given n small arrays, and you will create one big array from the concatenation of one or more instances of the small arrays (each small array could occur more than once). The big array will be given as an array of indexes (1-based) of the small arrays, and the concatenation should be done in the same order as in this array. Then you should apply the standard problem mentioned above on the resulting big array.

For example let's suppose that the small arrays are {1, 6, -2}, {3, 3} and {-5, 1}. And the indexes in the big array are {2, 3, 1, 3}. So the actual values in the big array after formatting it as concatenation of the small arrays will be {3, 3, -5, 1, 1, 6, -2, -5, 1}. In this example the maximum sum is 9.

Can you help Mostafa solve this problem?

Input

The first line contains two integers n and m, n is the number of the small arrays ($1 \le n \le 50$), and m is the number of indexes in the big array ($1 \le m \le 250000$). Then follow n lines, the i-th line starts with one integer l which is the size of the i-th array ($1 \le l \le 5000$), followed by l integers each one will be greater than or equal -1000 and less than or equal 1000. The last line contains m integers which are the indexes in the big array, and you should concatenate the small arrays in the same order, and each index will be greater than or equal to 1 and less than or equal to n.

The small arrays are numbered from 1 to n in the same order as given in the input. Some of the given small arrays may not be used in big array.

Note, that the array is very big. So if you try to build it straightforwardly, you will probably get time or/and memory limit exceeded.

Output

Print one line containing the maximum sum in the big array after formatting it as described above. You must choose at least one element for the sum, i. e. it cannot be empty.

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

Sample test(s)

input

```
3 4
3 1 6 -2
2 3 3
2 -5 1
2 3 1 3
```

output

```
9
```

input

```
6 1
4 0 8 -3 -10
8 3 -2 -5 10 8 -9 -5 -4
1 0
1 -3
3 -8 5 6
2 9 6
1
```

output

```
8
```

E. Ship's Shortest Path

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

You have got a new job, and it's very interesting, you are a ship captain. Your first task is to move your ship from one point to another point, and for sure you want to move it at the minimum cost.

And it's well known that the shortest distance between any 2 points is the length of the line segment between these 2 points. But unfortunately there is an island in the sea, so sometimes you won't be able to move your ship in the line segment between the 2 points.

You can **only** move to safe points. A point is called safe if it's on the line segment between the start and end points, or if it's on the island's edge.

But you are too lucky, you have got some clever and strong workers and they can help you in your trip, they can help you move the ship in the sea and they will take 1 Egyptian pound for each moving unit in the sea, and they can carry the ship (yes, they are very strong) and walk on the island and they will take 2 Egyptian pounds for each moving unit in the island. The money which you will give to them will be divided between all workers, so the number of workers does not matter here.

You can move your ship on the island edge, and it will be considered moving in the sea.

Now you have a sea map, and you have to decide what is the minimum cost for your trip.

Your starting point is ($xStart, yStart$), and the end point is ($xEnd, yEnd$), both points will be different.

The island will be a convex polygon and there will be no more than 2 polygon points on the same line, also the starting and the end points won't be inside or on the boundary of the island. The points for the polygon will be given in the anti-clockwise order.

Input

The first line contains 4 integers, $xStart, yStart, xEnd$ and $yEnd$ ($-100 \le xStart, yStart, xEnd, yEnd \le 100$). The second line contains an integer n, which is the number of points in the polygon ($3 \le n \le 30$), followed by a line containing n pairs of integers x and y, which are the coordinates of the points ($-100 \le x, y \le 100$), the polygon points will be distinct.

Output

Print one line which contains the minimum possible cost. The absolute or relative error in the answer should not exceed 10^{-6}.

Sample test(s)

input
1 7 6 7 4 4 2 4 12 3 12 3 2
output
6.000000000

input
-1 0 2 0 4 0 0 1 0 1 1 0 1
output
3.000000000