KTH Challenge 2016

Stockholm, 12th June 2016

Problems

This page is intentionally left (almost) blank.

Problem A
Ironman
Problem ID: ironman


Cropped from picture by JBLM MWR

An ironman triathlon is a race where participants swim for 3.86 km, ride a bicycle for 180.25 km, and finally run a marathon, and it is considered one of the toughest sport events. Viveka has been training for an even more challenging competition: the n-athlon. In an n-athlon race, participants have to go from the starting point to the finishing point through several types of terrain: water, sand, ice, asphalt, etc. To make the race more interesting, participants are free to pick the route that they think suits best their abilities. Last year Viveka achieved an epic victory by skating the last 40 km in 1 hour over ice, while her arch-rival Veronica was stuck in a tar pit 1 m from the finishing point.

The terrain distribution for this year has been published and now it is your task as the optimization expert in Viveka's team to help her figure out the best route for the race. The competition takes place in a flat area, which we model as the 2D plane, and each type of terrain is shaped as a horizontal strip. Participants are not allowed to leave the race area. You know the position of each strip and Viveka's speed in that type of terrain.

Input

The first line contains two pairs of decimal numbers x_s, y_s, x_f, y_f, the x and y coordinates of the starting and finishing point, respectively, in meters. The second line contains one integer n ($1 \leq n \leq 10\,000$), the number of layers. The third line contains $n-1$ decimal numbers, the y coordinate of each change between layers. Layers are given in order, this is, $y_s < y_1 < y_2 < \cdots < y_{n-1} < y_f$, so the shape of layer i is $(-10\,000, 10\,000) \times (y_{i-1}, y_i)$. The first and last layers extend only until the y coordinate of the starting and finishing point, this is they have shape $(-10\,000, 10\,000) \times (y_s, y_1)$ and $(-10\,000, 10\,000) \times (y_{n-1}, y_f)$ respectively. The fourth line contains n decimal numbers, Viveka's speed in each layer, in meters per second. All decimal numbers have absolute value at most 10^4 and at most 4 decimals.

Output

Output the minimum time required for Viveka to go from the starting to the finishing point. Your answer should be within absolute or relative error at most 10^{-6}.

Sample Input 1	Sample Output 1
0 0 0 100 2 50 5 1	60

Sample Input 2	Sample Output 2
0 0 100 100 2 50 1 5	71.5697725691

Problem B
Another Brick in the Wall
Problem ID: anotherbrick

The construction worker previously known as Lars has many bricks of height 1 and different lengths, and he is now trying to build a wall of width w and height h. Since the construction worker previously known as Lars knows that the subset sum problem is NP-hard, he does not try to optimize the placement but he just lays the bricks in the order they are in his pile and hopes for the best. First he places the bricks in the first layer, left to right; after the first layer is complete he moves to the second layer and completes it, and so on. He only lays bricks horizontally, without rotating them. If at some point he cannot place a brick and has to leave a layer incomplete, then he gets annoyed and leaves. It does not matter if he has bricks left over after he finishes.

Yesterday the construction worker previously known as Lars got really annoyed when he realized that he could not complete the wall only at the last layer, so he tore it down and asked you for help. Can you tell whether the construction worker previously known as Lars will complete the wall with the new pile of bricks he has today?

Input

The first line contains three integers h, w, n ($1 \leq h \leq 100$, $1 \leq w \leq 100$, $1 \leq n \leq 10\,000$), the height of the wall, the width of the wall, and the number of bricks respectively. The second line contains n integers x_i ($1 \leq x_i \leq 10$), the length of each brick.

Output

Output `YES` if the construction worker previously known as Lars will complete the wall, and `NO` otherwise.

Sample Input 1	Sample Output 1
2 10 7 5 5 5 5 5 5 5	YES

Sample Input 2	Sample Output 2
2 10 7 5 5 5 3 5 2 2	NO

This page is intentionally left (almost) blank.

Problem C
Zoning
Problem ID: zoning

A town is often divided into zones, e.g, industrial zones, commercial zones, and residential zones. If some residential zone is very far from all commercial zones, then the people living there will have a long journey whenever they want to do some shopping, and this is undesirable.

The input will consist of an $n \times n$ grid of square zones. Each zone is labeled `1` (residential), `2` (industrial), or `3` (commercial). When travelling from one zone to another you are allowed to move north, east, south or west, and the distance travelled is the number of zone boundaries you traverse. So the distance between two adjacent zones is 1, and the distance from the zone in square $(1, 1)$ (the most south-westerly zone) to the zone in square $(2, 3)$ is 3 (one step east and two steps north). You may never move off the grid.

Your task is to find the longest distance one has to travel from a residential zone to get to the commercial zone closest to that residential zone.

Input

The first line of input contains an integer n, $2 \leq n \leq 1500$, followed by n lines of length n giving the map of the city zones as an $n \times n$ matrix where each entry is `1`, `2`, or `3` depending on zone type. You can assume that the city has zones of all three types.

Output

Output a single integer d, the largest distance from a residential zone to its closest commercial zone.

Sample Input 1

```
4
1223
2123
2213
3212
```

Sample Output 1

```
3
```

Sample Input 2

```
2
12
33
```

Sample Output 2

```
1
```

This page is intentionally left (almost) blank.

Problem D
Dice Betting
Problem ID: dicebetting

Gunnar and his friends like games which involve rolling dice. Gunnar has a huge collection of 6-sided, 12-sided and 20-sided dice. All the games with dice started to bore him, so he came up with a new game. He rolls an s-sided die n times and wins if at least k different numbers appear in the n throws. An s-sided die contains s distinct numbers $1, \dots, s$ on its sides.

Since this is a game only for one person, Gunnar and his friends decided to make it more fun by letting other people bet on a particular game. Before you bet on a particular game, you would like to know how probable it is to throw at least k different numbers in n throws with an s-sided die. We assume that all numbers have the same probability of being thrown in each throw.

Input

The input consists of a single line with three integers n, s, and k ($1 \leq n \leq 10\,000, 1 \leq k \leq s \leq 500$). n is the number of throws, k the number of different numbers that are needed to win and s is the number of sides the die has.

Output

Output one line with the probability that a player throws at least k different numbers within n throws with an s-sided die. Your answer should be within absolute or relative error at most 10^{-7}.

Sample Input 1	Sample Output 1
3 3 2	0.888888889

Sample Input 2	Sample Output 2
3 3 3	0.222222222

This page is intentionally left (almost) blank.

Problem E
Climbing
Problem ID: climbing


Picture by Graham Evans

Fiona is an expert climber. She often brings some pegs with her, which she nails in some strategic places in the rock wall, so that less experienced climbers can use them for support. Fiona can climb to anywhere in the wall, but hammering a peg needs some balance, so she can only place a peg if she is standing in currently placed pegs (or, of course, the floor). She can remove a peg at any time and reuse it later. For each wall she is planning to visit, she has a careful plan for how to place and remove pegs in such a way that every strategic point has a peg at some step.

Yesterday it was raining, so the rock will be wet and it can be unsafe to remove pegs. Because of this, Fiona will only remove a peg p if she can stand on the same pegs as when p was placed. Alas Fiona's existing plans do not take this new precaution into account, so Fiona has to update her plans and she has asked you for help. She would like not to carry too many extra pegs, so you promised to find safe plans using at most 10 times more pegs than her previous plans were using. Can you deliver on your promise?

For example, consider the wall in the first sample input with 5 strategic points. Point 1 is close to the ground so it does not depend on any point. There has to be a peg in point 1 in order to put a peg in point 2, and the same holds for point 3. In order to put a peg in point 4, there has to be a peg both in point 2 and point 3. To put a peg in point 5 it is enough if there is a peg at point 4.

Therefore, the sequence (with annotations $+$ and $-$ depending on whether we add or remove a peg) $+1, +2, +3, -1, +4, -2, -3, +5$ is a safe dry plan, and it uses 3 pegs. However it is not a safe wet plan, because we remove the pegs at points 2 and 3 without support. The sequence $+1, +2, -2, +3, -1, +4, -3, +5$ only requires 2 pegs, but it is not safe at all because we add a peg to point 4 without there being a peg at point 2. The sequence $+1, +2, +3, -1, +4, +5$ is a safe wet plan, and it uses 4 pegs.

Input

The first line contains an integer n ($1 \leq n \leq 1\,000$), the number of strategic points in a wall. Each of the following n lines contains an integer p ($0 \leq p < n$) and a list of p integers. If line i contains $x_1, \ldots, x_p$ ($1 \leq x_j < i$), then all points x_j need to have a peg in order to place a peg in point i.

The next line contains an integer t ($1 \leq t \leq 1\,000$), the number of steps in the safe dry plan. Each of the following t lines contains an integer i, meaning that a peg is being placed or removed from point i.

Output

If there is no safe wet plan using at most 10 times the number of pegs of the safe dry plan, output -1. Otherwise, the first line must contain an integer t ($1 \leq t \leq 1\,000\,000$), the number of steps in the safe wet plan. Each of the next t lines must contain an integer i, meaning that a peg is being placed or removed from point i.

Sample Input 1

```
5
0
1 1
1 1
2 2 3
1 4
8
1
2
3
1
4
2
3
5
```

Sample Output 1

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

Sample Input 2

```
3
0
1 1
1 2
4
1
2
1
3
```

Sample Output 2

```
4
1
2
1
3
```

Problem F
Hay Bales
Problem ID: haybales


Picture by stanze

Peter has lined up hay bales. Some hay bales contain parasites and he wants to move the infected hay bales to the back of the sequence, to minimize the chance that the parasites spread. To sort the haybales, he repeatedly takes out any three consecutive hay bales and puts them back in sorted order. Your task is to calculate the minimum number of operations Peter has to execute to sort the sequence.

Input

The input contains a single string s ($3 \leq |s| \leq 500$), the sequence of hay bales. Each character of s is either 'C' (for a clean hay bale) or 'P' (for an infected one).

Output

The output must contain one integer, the minimum number of steps Peter has to execute.

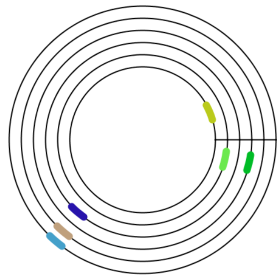
Sample Input 1	Sample Output 1
CPCC	1

Sample Input 2	Sample Output 2
PPPPCCCC	8

Sample Input 3	Sample Output 3
CCCCPPPP	0

This page is intentionally left (almost) blank.

Problem G
Racetrack
Problem ID: racetrack



There is a racetrack where n players complete laps. Each player has their own maximum speed. In this racetrack, overtaking is only possible near the finish line at every lap: when a player approaches a slower player, she will stay behind him until at the finish line. At the finish line, all players crossing the line at the same time resume driving at their maximum speed (so faster players overtake slower ones). Initially, all players start at the finish line. Given the lap time and the number of laps to complete for each player, calculate the times they complete the race in.

Input

The first line contains an integer n ($1 \leq n \leq 5\,000$), the number of players. The following n lines contain the players' lap time and number of laps to complete: the i-th line contains two integers t_i and c_i ($1 \leq t_i \leq 10^6$, $1 \leq c_i \leq 1\,000$), the lap time and the number of laps to complete for player i. The players are sorted in decreasing order of speed, that is, $t_1 \leq t_2 \leq \ldots \leq t_n$.

Output

Output n lines; the i'th line must contain the time that player i completes the race.

Sample Input 1	Sample Output 1
2 4 8 7 6	36 42

This page is intentionally left (almost) blank.

Problem H
nnnnn
Problem ID: nnnnn

Hsara and Simone like to communicate without anyone else knowing what they're saying. This time, Simone invented a very sneaky cipher. When she wants to tell Hsara a non-negative number n, she performs the following encryption procedure.

Let $d(n)$ denote the decimal expansion of n. Consider the string $x := d(n)^n$, i.e., the decimal expansion of n concatenated with itself n times. The encryption of n is then the length of x.

As an example, assume Simone wants to encrypt the number 10. Then

$$x = 10101010101010101010.$$

The length of x is then 20, which will be the encrypted value of x.

Hsara had no problem writing a decryption algorithm for this procedure. But can you?

Input

The first and only line contains an integer L ($0 \le L \le 10^{10^6}$), the encrypted value of some non-negative integer n.

Output

Output a single line containing the integer n.

Sample Input 1	Sample Output 1
20	10

This page is intentionally left (almost) blank.