Problem A: A Password Policy Requirement

Password security is a tricky issue. Users usually prefer short and simple passwords that are easy to remember, but such passwords are not secure enough. So, there should be some password policy requirements in security-enhanced environments. Below is an example of a password policy requirement for a user account in a domain:

The password should be at least six characters in length and should contain characters from all the following categories:

1) English uppercase letters (**A** through **Z**)
2) English lowercase letters (**a** through **z**)
3) Base 10 digits (**0** through **9**)

Given a string of alphanumeric (lower case, upper case, or digit) characters, your task is to find the length of its shortest contiguous substring which satisfies the above password policy requirement stated.

Input (Standard Input)

The input contains N test cases. The first line of the input has one integer N $(1 \leq N \leq 50)$.
Each of the next N lines is a test case having a string of at most 200 alphanumeric characters.

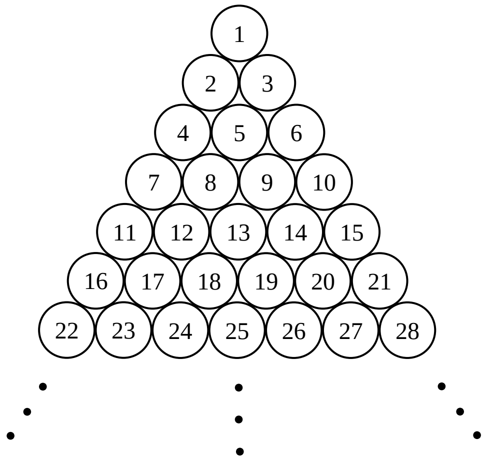
Output (Standard Output)

Write the result of the i^{th} test case, on the i^{th} line of output. You should just write one integer indicating the minimum length of a contiguous substring which satisfies the password policy. If there is no such substring, write "0".

Sample Input and Output

Standard Input	Standard Output
4 AliKam123test AbCdEfG 88syadneerG Windows7released21october2009	6 0 10 8

Problem B: Best Friends



Two friends, *Petey* and *Patty* are locked up in a maze. The maze has an infinite number of circles of the same size, arranged like the figure on the right. Petey and Patty are initially standing on two (not necessarily distinct) circles.

Petey wants to reach her friend Patty. In each step, she can go from the circle she is standing on, to one of the adjacent circles. Two circles are adjacent to each other, if they share a point.

Given the numbers (as shown in the figure) of the two circles Petey and Patty are standing on initially, you're to find the minimum number of steps Petey needs to reach her friend.

Input (Standard Input)

The input contains several test cases. Each test case is a line containing two space-separated integers specifying the initial circles Petey and Patty are standing on. None of these numbers is more than `10000`. The last line contains "`0 0`" which shows the end of the input, and should not be processed.

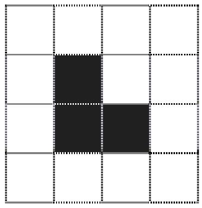
Output (Standard Output)

Write the result of the i^{th} test case, on the i^{th} line of output. You should just write one integer indicating the minimum number of steps Petey needs to reach her friend.

Sample Input and Output

Standard Input	Standard Output
1 3 2 6 23 9 0 0	1 2 4

Problem C: Calculate the Fence Needed



The Great Farmer has decided to build a fence around his farm. His farm is made up of some connected unit squares on a grid; the farm does not have any holes. The farmer needs to know the length of the fence required to surround his farm, and has asked for your help. Given the places of all the unit squares, your task is to calculate the perimeter of the farm. For example, in the figure on the right, the farm is made up of 3 (dark) unit squares, and its perimeter is 8.

Input (Standard Input)

There are multiple test cases in the input. Each test case starts with a line containing a single integer number N ($1 \leq N \leq 1000$), the area of the farm. Each of the next N lines has two space-separated integers x and y ($0 \leq x, y \leq 100$), where (x, y) shows the coordinates of the lower left corner of a unit square in the farm. The input terminates with a line containing "`0`" which should not be processed.

Output (Standard Output)

Write the result of the i^{th} test case, on the i^{th} line of output. You must write a single integer indicating the perimeter of the farm.

Sample Input and Output

Standard Input	Standard Output
3 1 1 1 2 2 1 4 3 3 3 4 4 4 4 3 4 1 2 1 3 1 4 2 4 0	8 8 10

Problem D. Diamonds

King *Terenas* ordered his minister to design a brain-stimulating game for prince *Arthas* – his son – to improve his mental capabilities. The minister designed a game in which there are N boxes labeled $1, \dots, N$; each box has a lock which can be opened only using its unique key. Before the game starts, the minister puts copies of the key of each box i in the boxes $i + 1$ and $i - 1$, if they exist. Afterwards, D diamonds are placed inside D distinct boxes. Finally, all of the boxes are locked and Arthas is given the keys to some M boxes. The Prince should collect all diamonds by opening the least number of boxes.

Arthas doesn't like too much thinking. "What are computers for, if we do the thinking?" he says. You are ordered by his majesty, prince Arthas, son of the mighty king Terenas, to make the computer think and solve this task.

Input (Standard Input)

There are multiple test cases in the input. The first line of each test case starts with the integer N $(1 \le N \le 500)$, the number of boxes, followed by M $(1 \le \boldsymbol{M} \le \boldsymbol{N})$, the number of keys given to Arthas, followed by D $(\boldsymbol{0} \le \boldsymbol{D} \le \boldsymbol{N})$, the number of diamonds. The M numbers on the next line represent the boxes whose keys are given to Arthas. Labels of the boxes containing the diamonds are listed on the next line. The input terminates with a line containing "`0 0 0`".

Output (Standard Output)

For each test case, write a single line containing the minimum number of boxes Arthas needs to open in order to collect all diamonds.

Sample Input and Output

Standard Input	Standard Output
5 1 5 1 1 2 3 4 5 5 3 3 1 2 3 3 4 5 5 3 1 1 2 3 5 0 0 0	5 3 3

Problem E. The Avaricious ISP

"Who can pay more money than the lonely, nevertheless affluent, internet user?" said the head of the Avaricious Council of Money-makers (ACM), as the concluding remarks of her talks.

As a prestigious member of the council, you feel so lucky to be also the head of an Internet Service Provider.
"Let's build a new network. Yeah! And … we can charge the users for the traffic. This cannot be better! I'm going to be rich." These are the thoughts that enter your mind immediately after the ACM head's talks.

Your network manager, a techie as you might guess, is not so good with economics. She doesn't understand anything about money. All she cares about is the so-called "topology of the network." But remember the last time you tried to argue with her. You don't want that to happen again; so just listen to her.

She insists on having two wireless antennas connected to each other by an underground cable. Each antenna can cover a circle of arbitrary radius, centered on the antenna. But, no point in the plane (yes, we're like those people who think the Earth is flat) should be covered by both antennas; it might be disastrous, your network manager suggests, as a device in the coverage of both antennas can be confused to death!

Don't worry though. There are some potential customers, located in some known points on the plane. Each will join your network if any of your antennas cover her. Each customer i has a known value d_i of traffic-desire. The amount of daily traffic between customers i and j would be equal to $d_i \times d_j$, provided that they are both covered by your network.

Because of the technical difficulties which your network manager points, you can only measure the amount of traffic passing through the underground cable. Hence, you should try to maximize the traffic passing through the cable; i.e., the traffic between pairs of users connected to different antennas. You can do that, or you can try to fire your network manager and devise a better plan. Of course, you are suggested to go with the first option.

Input (Standard Input)

There are multiple test cases in the input. The first line of each test case contains a single integer $1 \leq N \leq 200$, the number of potential customers. N lines follow, each containing the integer $1 \leq d_i \leq 200$ followed by a pair of integers $-10^6 \leq x_i, y_i \leq 10^6$ showing the coordinates of the potential customer. The input terminates with a line containing "0". For your convenience, the input data sets are designed so that no three potential customers of any given test lie on a common straight line.

Output (Standard Output)

For each test case, write a single line containing the maximum traffic passing through the underground cable, assuming you choose the best possible location and radius of coverage for each antenna.

Sample Input and Output

Standard Input	Standard Output
2 5 0 0 4 1 1 4 2 -100 -100 3 100 -100 4 -100 100 5 100 100 0	20 48

Problem F. Border Conflict

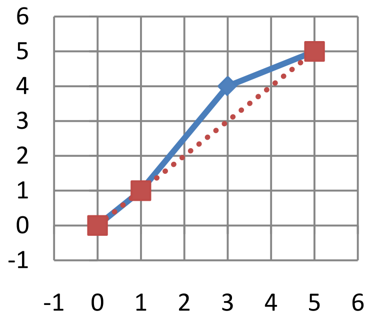
Irvanistan and Jikjikestan are two neighboring countries that have fought several wars with many casualties over their border dispute. Despite the loss of lives in the scale of tens of thousands, none of their border claims have been accepted by the other party.

Recently, the logical leaders who have gained control of both countries have accepted the United Nations proposal to resolve their border dispute. The proposal is to come up with a shorter and simplified version of the border that is calculated by a fair computer program.

To describe the problem accurately, let the current border P be a set of non-crossing line segments each connecting two border points. Let $p_0, p_1, \dots, p_N$ be these border points; i.e., P is exactly composed of the line segments connecting p_i and p_{i+1}, for $0 \le i < N$.

The UN suggests to create a new border C with points $c_0, c_1, \dots, c_K$, in such a way that $c_0 = p_0$ and $c_K = p_N$ and the following constraints are satisfied.

1. Each point c_i should be one of the points $p_0, \dots, p_N$. Obviously, if $c_i = p_r$ and $c_{i+1} = p_s$, then $s > r$.
2. Each point p_i should have a distance of at most some given number D from C. The distance of p_i from C is defined as the distance from p_i to the closest point on C. Note that, the line segment drawn from p_i to the closest point on C, is always perpendicular to C.



Your task is to find a new border C with the shortest possible length, while adhering to the above constraints.

Input (Standard Input)

There are multiple test cases in the input. The first line of each test case contains N ($2 \le N \le 100$), the number of points followed by D ($0 \le D \le 500$). Each of the next N lines contains two integers x_i, y_i ($-10000 \le x_i, y_i \le 10000$) which are the coordinates of the point p_i. Note that the point coordinates are increasing; i.e. $x_i < x_{i+1}$ and $y_i < y_{i+1}$. The input terminates with a line containing "0".

Output (Standard Output)

For each test case write a single line containing the shortest possible length of the new border with exactly two digits after the decimal point.

Sample Input and Output

Standard Input	Standard Output
4 3 1 1 2 2 4 5 6 6 3 0 1 1 5 4 8 8 0 0	7.07 10.00

Problem G: Goguryeo and the Crown Prince

The king Dongmyeong of Goguryeo, also known by his birth name Jumong, was the founding monarch of Goguryeo. Jumong had a son named Yuri from his first wife queen Yesoya, and two sons, Biryu and Onjo, from his second wife queen So Seo-no. They were happily living together, until it was the time to determine the true heir of Jumong. The candidates for the crown prince position are Yuri and Biryu.

In order to prevent a civil war, Jumong designed a game similar to Lotto and announced that the winner of the game would become the crown prince. In the beginning of the game, Jumong specifies an integer n, then each player chooses a binary string (a string of **`0`**'s and **`1`**'s) of length n. The selected strings must be different; if they are equal, the string selection step is repeated again.

Having received two different strings of the same length, Jumong starts tossing a fair coin (a coin with equal probabilities for heads and tails) for several times. Treating heads as **`0`**'s and tails as **`1`**'s, the sequence of coin flipping produces a binary string growing on the right. The player whose string appears earlier in the growing binary string, wins the game and becomes the crown prince.

Given the binary strings chosen by Yuri and Biryu, you have to calculate the winning probability of Yuri.

Input (Standard Input)

The input contains several test cases. Each test case is a line containing two space-separated binary strings chosen by Yuri and Biryu, respectively. The strings are different, but have the same length which does not exceed 30. The last line contains "`0 0`" which should not be processed.

Output (Standard Output)

Write the result of the i^{th} test case, on the i^{th} line of the output. You should only write one real number rounded to exactly 3 digits after the decimal point, which indicates the winning probability of Yuri.

Sample Input and Output

Standard Input	Standard Output
01 00 01 11 000 100 000 010 001 011 0 0	0.500 0.750 0.125 0.400 0.667

Problem H: Array Game

There is a single-player game played on a one-dimensional infinite-from-both-ends array containing integers, + signs and - signs. In each turn, the player can move all integers one cell to the left or one cell to the right (signs remain fixed).

The player's initial score is 0; when an integer I moves into a cell containing the sign S (+ or -), the integer is removed from the array and the score is increased by $S \times I$.

The player can stop the game anytime he/she wants.

Below you can see the initial and the following states of the array, after two *right moves* are made.

…	-1	0	1	2	3	4	5	6	7	8	9	10	…	
		2	**-**		5		**+**	-1			**+**			Score **=** 0
			-			5	**+**		-1		**+**			Score **=** -2
			-				**+**			-1	**+**			Score **=** 3

Your task is to find the maximum possible score one can get from a given initial array.

Input (Standard Input)

There are multiple test cases in the input. The first line of each test case starts with N $(1 \leq N \leq 100)$, the number of integers, followed by N_p $(1 \leq N_p \leq 100)$, the number of + signs and N_m $(1 \leq N_m \leq 100)$, the number of - signs. Each of the next N lines contains two integers p_i $(-300 \leq p_i \leq 300)$, the position and v_i $(-9 \leq v_i \leq 9)$, the value of the i^{th} integer. The following line contains N_p integers indicating the positions of the + signs. The following line contains N_m integers indicating the positions of the - signs. The positions are all between -300 and 300, and no two elements (integers and signs) are initially placed at the same position. The input terminates with a line containing "`0 0 0`".

Output (Standard Output)

For each test case write a single line containing the maximum possible score.

Sample Input and Output

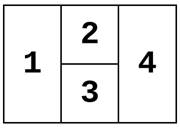
Standard Input	Standard Output
3 2 1 0 2 6 -1 3 5 5 9 1 1 1 1 10 5 3 7 0 0 0	3 0

Problem I: Tracking Robots

Several robots are moving around in an area, sending their locations to a server. Receiving a stream of locations sent by the robots, the server needs to find out the number of robots present in the area.

Assume that the area is a closed polygon, partitioned into some non-overlapping regions labeled $1, \ldots, N$. All robots are initially located in region 1. They all start moving around in the scene. When a robot enters a new region, it sends the region's label to the server. Note that each robot can enter and leave any region multiple times.

The server receives one long stream of region labels, without knowing the identity of the sender robots. Knowing the stream and the area's map, it needs your help to figure out the number of robots present in the area.



Your task is to find out the minimum and the maximum number of robots that may have created such a stream, assuming that each robot has at least once sent a region label to the server.

Input (Standard Input)

There are multiple test cases in the input. The first line of each test case starts with N $(1 \leq N \leq 100)$, the number of regions, followed by M $(1 \leq M \leq 200)$, the length of the server's stream. Each of the next N lines describes one region; the i^{th} line describes the region i. It starts with c_i, which is the number of regions adjacent to region i. c_i integers follow, indicating the labels of those regions. The last line is the server's stream which contains M region numbers, in the same order they were received by the server. The input terminates with a line containing "`0 0`".

Output (Standard Output)

For each test case write a single line containing the minimum and maximum possible number of robots present in the area.

Sample Input and Output

Standard Input	Standard Output
4 5 2 2 3 3 1 3 4 3 1 2 4 2 2 3 2 3 4 3 2 0 0	1 4

Problem J. ACM Revenge

Treasure hunters from all over the world have gathered together near the site of the Amazing Corridors of Mesopotamia (known as ACM among the treasure hunters).

"Over the years, many of us have entered ACM and never came back; now is the time for us to end this. We have gathered here to take revenge." These morale-raising words of the newly elected leader, do not affect you. You know better the purpose of the project *ACM Revenge*; this is all about the treasures hidden in ACM.

Data gathered from old Mesopotamian scripts provide the following insights:

1. ACM is a collection of rooms connected to each other via a number of *one-way* corridors. It is impossible to travel along a corridor in the wrong way.
2. There is an entrance room connected to the outside world. Each room in the ACM can be reached by a unique path from that entrance room.
3. Some rooms have no outgoing corridors. These rooms are filled with treasures, and as soon as one reaches them he/she and all the treasures inside will be teleported outside ACM, somewhere near the entrance.
4. Other rooms have exactly two outgoing corridors. At any given time, exactly one of these two corridors is blocked by a huge stone. As soon as someone enters the free corridor, the stone moves to block that corridor and frees the other one. In each of these rooms, there are a number of traps, each capable of killing one person. However each trap is used at most once and becomes inactive afterwards.

The new leader's plan is to send in treasure hunters one by one. As soon as one gets killed, his/her screaming is heard at the entrance; then another treasure hunter enters ACM. From the data you've collected, you know precisely what the map of ACM looks like, and how many working traps are still remaining in each room. You also know the corridors that are free right now. You want to be the first person who reaches one of the treasure rooms. You figure out that the m^{th} person who enters ACM reaches the first treasure room. Unfortunately, you need the help of a computer program to compute m.

Input (Standard Input)

There are multiple test cases in the input. The first line of each test case contains a single integer $1 \leq N \leq 20000$, the number of rooms in ACM. N lines follow, containing the descriptions of the 1st, 2nd, … and the N^{th} rooms. The i^{th} line contains three integers $0 \leq p_i \leq n$, $0 \leq f_i \leq 1$ and $0 \leq t_i \leq 100000$. p_i is the number of the room on the other end of the corridor leading to the i^{th} room; $p_i = 0$ means that the i^{th} room is the entrance. The number f_i is 1 if the corridor leading to the i^{th} room is currently free and is 0 otherwise. t_i shows the number of working traps located in the i^{th} room. The input terminates with a line containing 0. It's guaranteed that $f_i = 1$ for the entrance room, and that $t_i = 0$ for all rooms having no outgoing corridors. The input is terminated by a line containing "`0`".

Output (Standard Output)

For each test case, write a single line containing $m - 1$, the number of people who die before the first person reaches a treasure room.

Sample Input and Output

Standard Input	Standard Output
7 0 1 3 1 1 4 1 0 5 3 0 0 3 1 0 2 1 0 2 0 0 0	11