Contents

A. Airport Connections

For the purpose of this question, the phrases "airport route" and "airport connection" are used interchangeably. You're given a list of airports (three-letter codes like "JFK"), a list of routes (one-way flights from one airport to another like JFK, SFO), and a starting airport.

Write a program that returns the minimum number of airport connections (one-way flights) that need to be added in order for someone to be able to reach any airport in the list, starting at the starting airport.

Note that routes only allow you to fly in one direction; for instance, the route JFK, SFO only allows you to fly from "JFK" to "SFO".

Also note that the connections don’t have to be direct; it’s okay if an airport can only be reached from the starting airport by stopping at other airports first.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains comma-separated values the list of airports.

The second line of each test case contains a positive integer N which indicate the number of routes. And each N line contains comma-separated values which indicate the list of routes. And the last line of each test case contains the starting airport.

Output format:

returns the minimum number of airport connections that need to be added in order for someone to be able to reach any airport in the list, starting at the starting airport.

Constraints:

1 <= N <= 10^4

Time Limit: 3 seconds.

Example:

input	**Output**
1 BGI,CDG,DEL,DOH,DSM,EWR,EYW,HND,ICN,JFK,LGA,LHR,ORD,SAN,SFO,SIN,TLV,BUD 19 DSM,ORD ORD,BGI BGI,LGA SIN,CDG CDG,SIN CDG,BUD DEL,DOH DEL,CDG TLV,DEL	3

EWR,HND HND,ICN HND,JFK ICN,JFK JFK,LGA EYW,LHR LHR,SFO SFO,SAN SFO,DSM SAN,EYW LGA	

B. Max Profit with K Transactions

You're given an array of positive integers representing the prices of a single stock on various days (each index in the array represents a different day). You're also given an integer K, which represents the number of transactions you're allowed to make. One transaction consists of buying the stock on a given day and selling it on another, later day.

Write a program that returns the maximum profit that you can make by buying and selling the stock, given K transactions.

Note that you can only hold one share of the stock at a time; in other words, you can't buy more than one share of the stock on any given day, and you can't buy a share of the stock if you're still holding another share. Also, you don't need to use all K transactions that you're allowed.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains an integer K which represents the number of transactions you're allowed to make.

The second line of each test case contains comma-separated integer values representing the price of a single stock on various days.

Output format:

Returns an integer which is the maximum profit that you can make by buying and selling the stock, given K transactions.

Constraints:

1 <= K <= 100

Time Limit: 3 seconds.

Example:

Sample input	**Sample Output**
1 2 5,11,3,50,60,90	93

//output 93 = Buy: 5, Sell: 11; Buy: 3, Sell: 90

C. Same BSTs

An array of integers is said to represent the Binary Search Tree (BST) obtained by inserting each integer in the array, from left to right, into the BST.

Write a program that takes in two arrays of integers and determines whether these arrays represent the same BST.

A BST is a Binary Tree that consists only of BST nodes. A node is said to be a valid BST node if and only if it satisfies the BST property: its value is strictly greater than the values of every node to its left; its value is less than or equal to the values of every node to its right; and its children nodes are either valid BST nodes themselves or None / null.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains comma-separated values which indicate the first array.

The second line of each test case contains another comma-separated integer value representing the second array.

Output format:

Return true if the tow arrays represent the same BST or false otherwise.

Constraints:

1 <= array1.size <= 2000

1 <= array2.size <= 2000

Time Limit: 3 seconds.

Example:

Sample input	**Sample Output**
1 10,15,8,12,94,81,5,2,11 10,8,5,15,2,12,11,94,81	true

D. Pattern Matcher

You're given two non-empty strings. The first one is a pattern consisting of only "x"s and/or "y"s; the other one is a normal string of alphanumeric characters. Write a program that checks whether the normal string matches the pattern.

A string S_0 is said to match a pattern if replacing all "x"s in the pattern with some non-empty substring S_1 of S_0 and replacing all "y"s in the pattern with some non-empty substring S_2 of S_0 yields the same string S_0.

If the input string doesn't match the input pattern, the program should return an empty array; otherwise, it should return an array holding the strings S_1 and S_2 that represent "x" and "y" in the normal string, in that order. If the pattern doesn't contain any "x"s or "y"s, the respective letter should be represented by an empty string in the final array that you return.

You can assume that there will never be more than one pair of strings S_1 and S_2 that appropriately represent “x” and “y” in the normal string.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains a pattern string the “x”s and “y”s.

The second line of each test case contains another a normal string of alphanumeric characters.

Output format:

If the input string doesn't match the input pattern, the program should return an empty array; otherwise, it should return an array holding the strings S_1 and S_2 that represent "x" and "y" in the normal string, in that order

Constraints:

Time Limit: 3 seconds.

Example:

Sample input	Sample Output
1 xxyxxy gogopowerrangergogopowerranger	go powerranger

E. Circular game

Ahmed has invented the game as follows:

The participants throw a stone inside a circle to the center of the origin of the coordinates and with a radius of 5 meters, the score of each participant is calculated according to the distance of the stone from the center of the circle, and its value is the distance of the stone from the center of the circle. Each participant throws the stone into the circle twice and the score of this participant is the sum of the points. If the stone is placed outside the circle, the negative score is one, and if it is placed on the circle, it is zero. The circle is adjusted so that the stone is placed in a place with the integer coordinates.

Note: that the winner is the one with the lowest score.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains the number of participants

The second line of each test case contains the coordinates of the placement of the stone inside the circle

Output format:

Return score for each participant

Constraints:

Time Limit: 3 seconds.

Example:

Sample input	Sample Output
1 3 1,2 3,2 7,1 1,3 0,3 0,0	5.84162 2.16228 3

F. Longest Balanced SubString

Write a program that takes in a string made up of parentheses ((and)). The program should return an integer representing the length of the longest balanced substring with regards to parentheses.

A string is said to be balanced if it has as many opening parentheses as it has closing parentheses and if no parenthesis is unmatched. Note that an opening parenthesis can't match a closing parenthesis that comes before it, and similarly, a closing parenthesis can't match an opening parenthesis that comes after it.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains a string made up of parentheses.

Output format:

The program should return an integer representing the length of the longest balanced substring with regards to parentheses**.**

Constraints:

Time Limit: 3 seconds.

Example:

Sample input	**Sample Output**
1 (()))(	4

G. Red light

In a crowded city, a network of traffic cameras is installed on all crosses and capture the image of the cars that pass the red lights. The law enforcement department requires a software system to process all the captured numbers, and issue a ticket for all the cars that violate the law. An image processing module has already preprocessed the video recordings and generated a file containing the recorded car numbers. You are requested to write the ticketing module. The problem is that the image processing module is not perfect, and some car numbers are wrong.

A true car number should meet all of the following requirements:

- It is 8 characters long.
- The two leftmost characters are identical digits between 1 to 9, which indicate in which city the car number is issued.
- The following two characters are two digits between 1 to 9.
- The following character is a capital English letter.

- The three rightmost characters are also digits between 1 to 9.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains the number of captured car numbers (n) that are reported by the image processing module

In each of the next n lines, there are exactly 8 numerical or English alphabetical characters, showing one inferred car number.

Output format:

You should report the car numbers that have violated the law, one in each line, in the same order that they appear in the input. If the same car number has violated the law multiple times, all of the violation cases should be reported. The wrong car numbers, which do not meet one or more of the above requirements, should not be listed in the output.

Constraints:

You can assume $1 \leqslant n \leqslant 1000$.

Time Limit: 3 seconds.

Example:

Sample input	**Sample Output**
1 5 8835R551 4352S132 2241X223 55123456 9914t521	8835R551 2241X223

H. Largest Rectangle Under Skyline

Write a program that takes in an array of positive integers representing the heights of adjacent buildings and returns the area of the largest rectangle that can be created by any number of adjacent buildings, including just one building. Note that all buildings have the same width of 1 unit.

For example, given `buildings = [2, 1, 2]`, the area of the largest rectangle that can be created is 3, using all three buildings. Since the minimum height of the three buildings is 1, you can create a rectangle that has a height of 1 and a width of 3 (the number of buildings). You could also create rectangles of area 2 by using only the first building or the last building, but these clearly wouldn't be the largest rectangles. Similarly, you could create rectangles of area 2 by using the first and second building or the second and third building.

To clarify, the width of a created rectangle is the number of buildings used to create the rectangle, and its height is the height of the smallest building used to create it.

Note that if no rectangles can be created, your program should return 0.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains comma-separated values representing the heights of adjacent .

Output format:

returns the area of the largest rectangle that can be created by any number of adjacent buildings, including just one building if no rectangles can be created, your program should return 0.

Sample input	**Sample Output**
1 1,3,3,2,4,1,5,3,2	9

```
// below is visual representation of the sample input.
//             _
//         _  | |
//   _ _  | | | |_
//  | | |_| | | | |_
// _| | | | |_| | | |
//|_|_|_|_|_|_|_|_|_|
```

I. Tax

Every employee, in every office, must pay taxes at the end of the month. In Afghanistan, the tax is deducted from the monthly income as follows:

- There is no tax for less than 5000 Afghanis.
- 5% is deducted up to 12,000 Afghanis.
- 10% is deducted up to 100,000 Afghani.
- More than 100,000 Afghanis will be deducted 20%.

The financial manager of Kateb University needs a system that, by receiving the number of employees as well as their monthly salaries, calculates their taxes and deducts from the basic salary and prints the salary paid to the employee.

Note: Someone who has a monthly salary of 13,000 afghanis does not pay tax up to 5,000 afghanis, for 7,000 from 5,000 to 12,000, 5% of it and for the remaining 1,000, 10% tax is calculated.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains is the N which is number of employees.

In each next N line is of monthly salary of employees.

Output format:

returns the Salaries of employees after deduction of taxes

Sample input	**Sample Output**
1 3 6500 14000 25000	6425 13450 23350

J. Ambiguous Measurements

This problem deals with measuring cups that are missing important measuring labels. Specifically, a measuring cup only has two measuring lines, a Low (L) line and a High (H) line. This means that these cups can't precisely measure and can only guarantee that the substance poured into them will be between the L and H line. For example, you might have a measuring cup that has a Low line at 400ml and a high line at 435ml. This means that when you use this measuring cup, you can only be sure that what you're measuring is between 400ml and 435ml.

You're given a list of measuring cups containing their low and high lines as well as one Low integer and one High integer representing a range for a target measurement. Write a program that returns a boolean representing whether you can use the cups to accurately measure a volume in the specified [low, high] range (the range is inclusive).

Note that:

- Each measuring cup will be represented by a pair of positive integers [L,H], where 0≤L≤H.
- You'll always be given at least one measuring cup, and the low and high input parameters will always satisfy the following constraint: 0≤low≤high.
- Once you've measured some liquid, it will immediately be transferred to a larger bowl that will eventually (possibly) contain the target measurement.
- You can't pour the contents of one measuring cup into another cup.

Input format:

The input consists of multiple test cases so the first line of the input is the number of test cases.

The first line of each test case contains is the N which is number the number of list of measuring cups.

In each next N line contains two comma-separated values which indicate the list of measuring cups containing their low and high lines.

The next line is Low integer

And the last line is high integer

Output format:

returns a boolean representing whether you can use the cups to accurately measure a volume in the specified [low, high] range (the range is inclusive)

Sample input	**Sample Output**
1 3 200,210 450,465 800,850 2100 2300	true