The 2023 ICPC Asia Kabul

A. Waterfall Streams

You're given a two-dimensional array that represents the structure of an indoor waterfall and a positive Integer that represents the column that the waterfalls water source will start at. More specifically, the Water source will start directly above the structure and will flow downwards.

Each row in the array contains Os and 1s, where a 0 represents a free space and a 1 represents a block That water can't pass through. You can imagine that the last row of the array will always contain only Os.

You can also imagine that there are walls on both sides of the structure, meaning that water will never Leave the structure; it will either be trapped against a wall or flow into one of the buckets in the last row.

As water flows downwards, if it hits a block, it splits evenly to the left and right-hand side of that block. In other words, 50% of the water flows left and 50% of it flows right, if a water stream is unable to flow To the left or to the right (because of a block or a wall), the water stream in question becomes trapped And can no longer continue to flow in that direction; it effectively gets stuck in the structure and can no Longer flow downwards, meaning that 50% of the previous water stream is forever lost.

Lastly, the input array will always contain at least two rows and one column, and the space directly Below the water source (in the first row of the array) will always be empty, allowing the water to start Flowing downwards.

Your task is to write a code that return the percentage of water inside each of the bottom buckets after The water has flowed through the entire structure.

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 positive integer that represents the column that the waterfalls Water source will start at.

The second line of each test case contains two space-separated integers R denoting the number of rows And C denoting the number of columns in the two-dimensional array. Next R line contains the data for Two each row of the two-dimensional array separated by a space.

Output format:

Your objective is to output the percentage of water inside each of the bottom buckets after the water

Has flowed through the entire structure.

Constraints:

1 <= R <= 10^4

1 <= C <= 10^5

Time Limit: 3 seconds.

Example:

Sample input	**Sample Output**
1 3 7 7 0000000 1000000 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	0.0 0.0 0.0 25.0 25.0 0.00.0

The water will flow as follows:

0 0 0 . 0 0 0

1 . . . . . .

0 . 1 1 1 . 0

. . . . . .

1 1 1 . . 10

0 0 0 . . 0 1

0 0 0 . . 0 0

B. Count Inversions

Your task is to write a program that takes in an array of integers and returns the number of inversions in

The array. An inversion occurs if for any valid indicesi and j, i<j and array [i]> array[j].

For example, given array = 43, 4, 1, 2], there are 4 inversions. The following pairs of indices represent

Inversions: [0, 2], [0, 3], [1, 2], [1,3].

Intuitively, the number of inversions is a measure of how unsorted the array is.

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 N (1 <= N <= 10^4), which represents the length of the

Array. The second line contains n integers separated by a space, the elements of the input array.

Output format:

For each test case, print a single integer, which is the number of inversions.

Time limit: 3 secs

Example:

Sample input	**Sample Output**
1 5 8 19 320	9

C. Palindrome Partitioning Min Cuts

Given a non-empty string, write a program that returns the minimum number of cuts needed to perform

On the string such that each remaining substring is a palindrome.

A palindrome is defined as a string that's written the same forward as backward. Note that single-

Character strings are palindromes.

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 non-empty string in which you have find the minimum number of

Cuts

Output format:

For each test case, print a single integer, which is the number of cuts needed to perform on the string

Such that each remaining substring is a palindrome.

Time limit: 3 secs

Example:

Sample input	**Sample Output**
1 Noonabbad	2

First Text case explanation: noon | abba | d

D. laptop Rentals

You're given a list of time intervals during which students at a school need a laptop. Theses time Intervals are represented by pairs of integers (start, end], where 0 <= start < end. However, start and End don't represent real times; therefore, they may be greater than 24.
No two students can use a laptop at the same time, but immediately after a student is done using a Laptop, another student can use that same laptop. For example, if one student rents a laptop during the Time interval [0, 2], another student can rent the same laptop during any time interval starting with 2. Write a program that returns the minimum number of laptops that the school needs to rent such that all Students will always have access to a laptop when they need one.

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 positive integer N which represents the number of time

Intervals.

Each N lines must contain two space-separated integers 5 denoting the start and E denoting the end of

The time interval.

Output format:

For each test case, print a single integer which is the minimum number of laptops that the school needs

To rent such that all students will always have access to a laptop when they need one.

Time limit: 2 secs

Constrains:

1 <= N => 1000

1 <= S=> 1000

1 <= E=> 1000

Example:

Sample input	**Sample Output**
1 7 0 2 4 6 0 4 7 8 9 11 3 10	3

E. Tandem Bicycle

A tandem bicycle is a bicycle that's operated by two people: Person A and person B. Both people pedal The bicycle, but the person that pedals faster dictates the speed of the bicycle. So if person A pedals at a Speed of 5, and person B pedals at a speed of 4, the tandem bicycle moves at a speed of 5 (i.e., tandemSpeed= max(speedA, speedB)).

Your given two lists of positive integers: one that contains the speeds of riders wearing red shirts and One that contains the speed of riders wearing blue shirts. Each rider is represented by a single positive Integer, which is the speed that the rider pedals a tandem bicycle at. Both lists have the same length, Meaning that there are as many red-shirt riders as there are blue-shirt riders. Your goal is to pair every Rider wearing a red shirt with a rider wearing a blue shirt to operate a tandem bicycle.

Write a program that returns the maximum possible total speed or the minimum possible total speed of All of the tandem bicycles being ridden based on an input parameter, fastest. If fastest = true, your Program should return the maximum possible total speed; otherwise it should return the minimum total Speed.

"Total speed" is defined as the sum of the speeds of all the tandem bicycles being ridden. For example, if There are 4 riders (2 red-shirt riders and 2 blue-shirt riders) who have speeds of 1, 3, 4, 5. And if they're Paired on tandem bicycles as follows: [1, 4], [5, 3), then the total speed of their tandem bicycles is 4 + 5 = 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 a line of comma-separated values representing red shirt speeds.

The second line of each test case contains another line of comma-separated values representing blue Shirt speeds.

The last line contains either True/true or False/false that indicates fastest parameter.

Output format:

Output the maximum possible total speed or the minimum possible total speed of all of the tandem

Bicycles being ridden based on an input parameter, fastest. If fastest = true, your program should return

The maximum possible total speed; otherwise it should return the minimum total speed.

Time limit: 3 sec

Example:

Sample input	**Sample Output**
1 5,5,3,9,2 3,6,7,2,1 True	32

F. Number Of Ways To Traverse Graph

You're given tow positive integers representing the width and height of a grid-shaped, rectangular Graph. Write a program that returns the number of ways to reach the bottom right corner of the Graph when starting at the top left corner. Each move you take must either go down or right. In Other words, you can never move up or left in the graph.

For example, given the graph illustrated below, with width=2 and height = 3, there are three ways To reach the bottom right corner when starting at the top left corner:

LLI

LLI

1. Down, Down, Right

2. Right, Down, Down

3. Down, Right, Down

Note: you may assume that width * height >= 2. In other words, the graph will never be a 1 * 1 grid.

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 two space-separated integers W denoting the width of

Rectangular graph and H denoting the height of rectangular graph.

Output format:

Your code should return the number of ways to reach the bottom right corner of the graph when

Starting at the top left corner.

Time limit: 3 sec

Constraints:

1 <= W => 105

1 <= H=> 105

Example:

Sample input	**Sample Output**
1 4 3	10

G. Number Of Binary Tree Topologies

Write a program that takes in a non-negative integer N and returns the number of possible Binary Tree topologies that can be created with exactly N nodes.
A Binary Tree topology is defined as any Binary Tree configuration. Irrespective of node values. For Instance, there exist only two Binary Tree topologies when n is equal to 2: a root node with a left Node, and a root node with a right node.
Note that when N is equal to 0. There's one topology that can be created: The None / null node.
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 positive integer N which represents the number of possible

Binary Tree.

Output format:

For each test case, print the number of possible Binary Tree topologies that can be created with

Exactly N nodes.

Time limit: 3 sec

Constraints:

1 <= N = 1000

Example:

Sample input	**Sample Output**
1 3	5

H. Topological Sort

You're given a list of arbitrary jobs that need to be completed; these jobs are represented by distinct Integers. You're also given a list of dependencies. A dependency is represented as a pair of jobs Where the first job is pre-requisite of the second one. In other words, the second job depends on the First one; it can only be completed once the first job is completed.

Write a program that takes in a list of jobs and a list of dependencies and return a list containing a Valid order in which the given jobs can be completed. If no such order exists, the program should Return Impossible.

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 positive integer N that represents the number of job and Job's dependency

The following N lines contain two space-separated integers J and D, which J is denoting the job and D Is the job dependency.

The next line contains a comma-separate value of distinct integers representing the job list.

Output format:

Your program should return a list of comma-separate value which is a valid order in which the given Jobs can be completed, if no such order exists, the program should return Impossible.

Time limit: 3 sec

Constraints:

1<= N => 1000

Example:

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

I. Levenshtien Distance

Write a program that takes in tow strings and returns the minimum number of edit operations that Need to be performed on the first string to obtain the second string.
There are three edit operations:
Insertion of character
Deletion of a character
And substitution of a character for another.
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 non-empty string representing first string.
The second line of each test case contains a non-empty string representing second string.
Output format:

Return an integer which is the minimum number of edit operations needed to be performed on the

First string to obtain the second string.

Time limit: 3 sec

Example:

Sample input	**Sample Output**
1 Abc yabd	2

First text case explanation: insert “y”: substitute “c” for “d”

J. Nth Fibonacci

Constraints:

$1 <= N => 9^{10}$

Simple Input	Simple Output
1 2	1

K. BESTFRIEND

One of your best friends is a database specialist in a university and they assigned him a task to
Develop a small application for managing a group of students' results in a single semester for various
Otherwise he will lose his job. Please take the following Credit System rules of Afghanistan
Subjects. He cannot accomplish the task and wants you to help him by this task to save his job
Universities into consideration:
Every Course has a prespecified value (credit).
Students are scored between 0 to 100 and also it is possible to get decimal points.
Students who get lower than 55 are calculated as failed in that particular course.
Students who fail in more than 50% of the credits are failed in that semester. Please notice
That if the total credit is odd and students fail in
Total of credits
2
+0.5 credit they are
Considered as passed in that semester. For example, if total credit is 19 and students fail in
10 credits they are passed.
In grading the students with lower number of failed credits and higher total percentage are
Ranked in the top. Additionally, if two or more students have the same failed credit number
And percentage sort in ascending alphabet order of their names.

Input format:

The first line contains the number of the test cases (T).
The next line contains the number of courses © for current test case.
U
The next line contains the courses name (CN) and its associated credit ® which are
Separated by colons and commas.
0-100

Next line contains the number of students (S).
Next S lines contains ID number, name (N), and marks (M) for each student (in the same Order of course-credit list) separated by commas.
The last line contains the ID of the requested students separated by commas “,” as a string And you have to print out their academic information according to output format.

Output format:

Print the list of requested students sorted in descending order- the top student in top-
Using the following format:
Std-name, std-ranking, total-percentage (up to 2 decimal digits), num-of-failed-courses, Failed-semester-or-no?
Example: $3,1,100,0,false
After printing each test case’s results print out ten (-) at the end.
Example:

Input	Output
2 5 English:3, Mathematics:4,00P:4,Sport:2,Database:4 4 1,Wali,80,95,86,90,97 2,Bashir,96,98,50,99,92 3, Halim,84,76,88,90,64 4,Omid, 100,81,85,94,95 2,1 2 A:4,B:2 5 1,S1,80,56 2,S5,100,0 3,S3,100,100 4,S4,56,89 5,52,49,92 3,1,5	Wali,2,85.11,0,false Bashir,4,80.33,1,false …… S3,1,100,0,false S1,2,72,0,false S2,5,63.33,1,false ……