The 2024 ICPC Asia Topi Regional Preliminary On-line Contest (MOCK ROUND)

Instructions

- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- If you have any question regarding the problems, send a clarification from the judges using DOMJudge.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3
- Source code file name should not contain white space or special characters.
- You must take input from Console i.e.: Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console i.e.: Standard Output Stream (stdout in C, cout in C++, System.out in Java)
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity etc.**
- Unless mentioned in some problem, all your programs must meet the time constraint of 5 seconds.
- The decision of judges will be absolutely final.

Problem 0: Hello World:

Write a program to print “Hello World!”.

Problem 1: Wedding Shopping

Time limit: 3 seconds

If there are different models of each item of clothing (for example, three shirts, two belts, four shoes, etc.), you should buy one model of each item of clothing. Unfortunately, we can't spend more than the budget allows, but we do want to spend as much as we can. Because of our limited funds, we might not be able to purchase one model of each item of clothing. That's followed by information about the C clothes. If you give us a garment ID between 0 and C-1, we can tell you how many different models that ID has, with K integers showing the prices of each model.

Input

In the very first line, you'll see the total number of test cases. For each test case, the total budget amount, and the number garment classes ***M*** and ***C***, comma separated are taken from the second line of the input file. Subsequently the ***C*** pair of lines will represent the number of garments and price for each garment.

Output

The output should consist of one integer at each line as per number of test cases that indicates the maximum amount of money necessary to buy one element of each garment without exceeding the initial amount of money. If there is no solution, print “-1”.

Sample input & output

The following is an example of a sample input and corresponding correct outputs.

Sample input	Sample Output
2 10 3 2 4 5 3 2 3 4 2 1 2 20 3 3 6 4 8 2 5 10 4 1 5 3 5	10 19

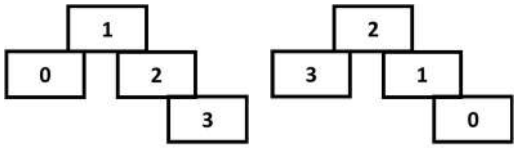
Problem 2: Architectural Dilemma

Time limit: 3 seconds

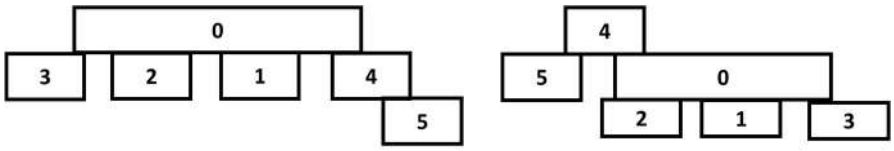
ACM (All components' merchants) is a well-known shopping mall chain. They want to create a new shopping plaza with several shops and showrooms. However, based on their experiences, they noticed that it is a good idea to pave the path of visitors through different shops. That increases the chances of sales of products and hence increase in revenue. According to their survey department, customers should start from the top and make their way to the lower levels moving through various shops based on their interest. However, shops at one level are not connected with each other. Moreover, as they expect a huge crowd, they want to distribute the crowd in different shops and let them out as soon as possible to make space for new customers.

They hired an architect to design the building for them. However, when they provided the constraints, the architect got puzzled and sought your assistance to devise a mechanism to generate a plan. After detailed discussion, ACM agreed to provide the total number of shops, ***n (1 <= n <= 1000),*** where shop numbers always start with ***0***, and possible connectivity among shops ***(n-1)***. The customer can move in either direction if two shops are connected. Thus, order of connectivity does not matter. The goal is to determine the number of possible ways which can make the plaza smallest in height while fulfilling the constraints.

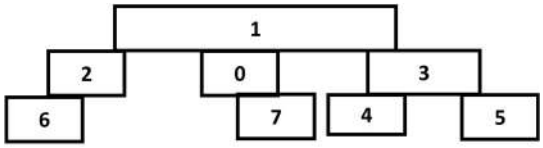
Example 1:
n = 4
Connections = [1, 0], [1, 2], [2, 3]
Output = 2
As following are two possibilities to maintain smallest height.



Example 2:
n = 6
Connections = [1, 0], [0, 2], [3, 0], [4, 0], [5,4]
Output = 2
As following are two possibilities to maintain smallest height.



Example 3:
n = 8
Connections = [1, 0], [1, 2], [3, 1], [6, 2], [3, 4], [3, 5], [7, 0]
Output = 1



Input

The first line of the input consists of t, $(1 \leq t \leq 25)$ representing the total number of test cases given. The first line of the test case contains number of shops as n, $(1 \leq n \leq 10^3)$, next $n-1$ lines contain space separated connected shop numbers.

Output

Output consists of t lines, having possible ways of building the plaza. If no solution exists, print -1.

Sample input	Sample Output
4 4 1 0 1 2 2 3 6 0 1 2 0 0 3 4 0 4 5 8 1 0 1 2 3 1 6 2 3 4 3 5 7 0 3 0 1 1 2	2 2 1 1

Problem 3: Network Traffic Monitoring

Time limit: 3 seconds

A university wants to make its internet robust against disconnections, network load and loss of traffic. For that they want to install a system for network traffic monitoring that consists of a sniffer that count the number of packets received per second, and a traffic monitor that shall raise an alarm when there is an anomaly. The network engineers decided that the monitor would raise an alarm when the absolute value of second-order difference between counts of packets is above a certain threshold.

Let the sniffer counts network packets received each second (which is an integer) xi such ***as 0≤xi≤1000***. The first-order difference is $\boldsymbol{\Delta_i = |x_i - x_{i-1}|}$ and second-order difference is given by $\boldsymbol{\Delta_i^2 = |\Delta_i - \Delta_{i-1}|}$. Let's denote the threshold by $\boldsymbol{\theta}$. Given a list of n integers xi where ***1≤n≤109*** and an integer threshold $\boldsymbol{\theta}$ where $\boldsymbol{1 \leq \theta \leq 1000}$, if $\boldsymbol{\Delta^2 i \geq \theta}$ then your program should print a "1", else print "0".

Input

The first line of input is the number of test cases, $\boldsymbol{t}$, the second line is the threshold $\boldsymbol{\theta}$. From third line onwards the test cases will be input. Such as in the example given below, he third line contains integer $\boldsymbol{n}$, number of packets per second in first case, followed in next line by list of $\boldsymbol{n}$ integers $\boldsymbol{x_i}$ for $\boldsymbol{i} \in \{\boldsymbol{1}, \ldots, \boldsymbol{n}\}$, space separated. Similarly, the fifth line is the length $\boldsymbol{n}$ of second test case, followed by another list for $\boldsymbol{x_i}$. Consequently, each two next lines contains $\boldsymbol{n}$ and list of $\boldsymbol{x_i}$.

Output

A list of integers for each test case of list xi. For each $\boldsymbol{x_i, i \geq 3}$ in a test case, print "0" if $\boldsymbol{\Delta_i^2 < \theta}$ and "1" if $\boldsymbol{\Delta_i^2 \geq \theta}$.

Sample Input	Sample Output
2 8 7 3 2 3 17 12 4 20 5 19 20 21 18 5	0 1 1 0 1 0 0 1

Problem 4: Jagga Tax

Time limit: 3 seconds

A term "Jagga Tax" is very common in our society. Chaudhry Muhammad Sharif Gujjar alias Jagga Gujjar of Islamia Park Chowburji Lahore, glorified in underworld's history for imposing the notorious 'Jagga Tax' on city's Bakar Mandi in the late 1960s, was also among the earliest casualties of these encounters under the 1959 Goonda Act. Legend has it that Bakar Mandi's butchers would pay Jagga Gujjar one rupee on every goat to be slaughtered.

You are required to help a gangster of your generation to place his extortion operations. In a region, there are ***N*** different crossings connected by a network of ***M*** roads. A group of ghundas plan to establish tax collection points on 5 of the crossings to maximize their revenue. To decide the optimal crossings for their operations, they consider the concept of "Jagga Revenue Potential" (JRP).

The JRP of a crossing is calculated as the number of shortest routes between all other crossings that pass through it. In other words, it measures how strategically located a crossing is for efficient tax collection. Write a program to help the bandits determine the top five crossings where they should establish the tax collection point to maximize their revenue, as measured by the JRP.

Input:

The first line contains an integer indicating the number of test cases. The second line contains two space separated integers, $N\ (2 \leq N \leq 1000)$ and $M\ (1 \leq M \leq 5000)$, representing the number of crossings and roads in the region, respectively. Consequently, the next ***M*** lines contain two integers each, U_i and $V_i\ (1 \leq U_i, V_i \leq N)$, indicating a road stretch that connects crossings U_i and V_i.

Output:

Output 5 integers, the index of the top 5 crossings where the jagga collection point should be established to maximize the bandits' revenue. If there are multiple valid solutions, output the one with the smallest index.

Sample Input	Sample Output
1 6 6 1 2 1 3 2 4 3 4 4 5 4 6	4 2 3 1 5

Problem 5: The closely link birds – Avian Breed Discrimination via Protein Sequencing

Time limit: 3 seconds

Numerous chicken breeds are popular in poultry farming due to their valuable traits such as high egg production (around 340 eggs annually), large egg size (approximately 30 cm diameter), vibrant plumage for ornamental purposes, and fast meat production. In contrast, ducks are generally considered a less profitable investment due to slower growth, drab appearance, and minimal egg laying (only ten eggs per year). Ducks also demand substantial water resources and hygiene maintenance to prevent disease transmission.

The Pakistan Finance Ministry aims to boost the country's economy by investing in chickens and has developed a method to distinguish between chicken breeds and ducks. Chickens share common protein sequences, which are strings of characters ranging from 3 to 20 characters in length. A protein sequencing machine analyzes these sequences and compares them. If two sequences are identical (e.g., ABCKD – ABCKD) or have equivalent amino acid matches based on a provided table, they are considered equivalent proteins (e.g., given amino acid 'A' matches with amino acid 'K', protein sequences 'ABCKD' and 'ABCAD' are equivalent).

The Finance Ministry reached out to the Pakistan Poultry Association to differentiate between chicken breeds and ducks using protein sequencing data. The association possesses a database of equivalent amino acids and a list of proteins from chickens and their breeds. The protein sequencing machine processes protein samples and matches the resulting protein sequence against stored chicken sequences. If a match is found, the specimen is identified as a chicken breed; otherwise, it is not classified as belonging to any chicken breed. A typical chicken protein sequence comprises 26 amino acids represented by English alphabets.

Given a list of equivalent amino acids and a set of proteins, the task is to determine if pairs of proteins are equivalent. Equivalence is established if the two proteins have the same length, and the amino acids in the first protein can be converted into corresponding amino acids in the second protein using the provided equivalence rule at least once.

For example, in the first test case (fourth example) given below, the strings are FFO and SHT. When finding equivalence between both strings, F translates to S directly (F→S), F translates to H via S (F→S→H), and finally O translates to T via C (O→C→T). Hence, FFO is equivalent to SHT, and the output is Yes.

Input

The input file is organized as follows: The first two lines of each test case represent the mapping. The first of these lines contain space- separated amino acid letters (Y_1 Y_2 Y_3 ... Y_n) and the second line contains the equivalent of each amino acid of line 1 separated by space (Z_1 Z_2 Z_3 ... Z_n). The remaining lines contain a pair of strings to match, separated by a space.

Output

Your output should have as many lines as the number of strings to match (of all test cases). For each pair of strings, display Yes if the two strings are equivalent and valid, and no otherwise.

Note: Amino acids and protein pairs use only uppercase letters 'A'– 'Z', and each protein sequence contains **at least 3** and **at most 20 letters**.

Sample Input 1	Sample Output 2
C I K O R T T S W F	Yes
T R P C O E F H P S	No
KROC KROC	No
NAC NOC	Yes
STTKCAR STEKCOR	Yes
FFO SHT	
SKOOITSF SKCOREHS	

Sample Input 2	Sample Output 2
A B A C	Yes
C A B A	No
AAA ABC	Yes
ABCD AAAA	No
ACM BCM	Yes
AC BC	No
ACABAACCBBCCAA BBBACCBBCCABBC	
ABACABACABACABACABACABACA ABACABACABACABACABACABACA	

Problem 6: Efficient HR

Time limit: 3 seconds

A Human Resource (HR) department of a company has divided its employees in departments; however, an employee can be a part of multiple departments. There is ***m (1 < m <= 100)*** departments which are uniquely identified with their id's ***(m)***. In each department we can have different number of people ***(n)*** where ***1 <= n <= 10***. The company want to send their employees for training sessions, but due to financial constraints they need to send employees from a unique department, in available seats. So, count the total number of arrangements or ways such that all person belongs from unique departments.

Input

The first line is indicating the number of test cases. The first line contains the number of people to be sent for training. The next line shows the number of departments a person belongs, followed by the department IDs in the next line. Similarly, the next lines will contain the data for remaining people.

Output:

It shows all the number of all possible unique combinations. In case there is no unique combinations the output will show 0.

Sample input	Sample Output
1 3 3 5 100 1 1 2 2 5 100	4