ACM ICPC Malaysia al Khawarizmi National Programming Contest 2014

CONTEST

This problem set contains 10 questions (A – J) in 21 pages.

28th September, 2014

Jointly organized by

Department of Computer Science and Information Sciences

Universiti Teknologi PETRONAS

&

Kulliyyah of Information and Communication Technology,

International Islamic University Malaysia

as the ACM ICPC Malaysia Office

A	KEEP IT SECRET & SAFE
	Input: Standard Input
	Output: Standard Output
	Time Limit: 10 seconds

Problem Description

Choosing the right password is something that some people find difficult. There are so many software applications that require password these days. Remembering all the chosen password can also be a real problem. Perhaps because of this a lot of people choose their password very badly, as long as they can remember.

A simple tip to choose a password is as follows:

- Use at least eight characters and keep it less than 15, because anything more than 15 character is difficult to remember

AND

- Use at least one character from both category below:
    a) letters (A – Z, a – z)
    b) digits (0 – 9) or symbols (all printable symbol on a basic keyboard, not including space `' '`)

Write a program that will determine whether a password is acceptable or not.

Input

First line of the input contains ***T*** the number of test cases (1 <= ***T*** <= 100). Each test case consists of a string ***S*** which represents a password. The maximum length of string ***S*** is 15 characters.

Output

For each test case, the output contains a line in the format Case #x: ***M***, where x is the case number (starting from 1) and ***M*** is either "Acceptable" or "Not acceptable"

Sample Input Output

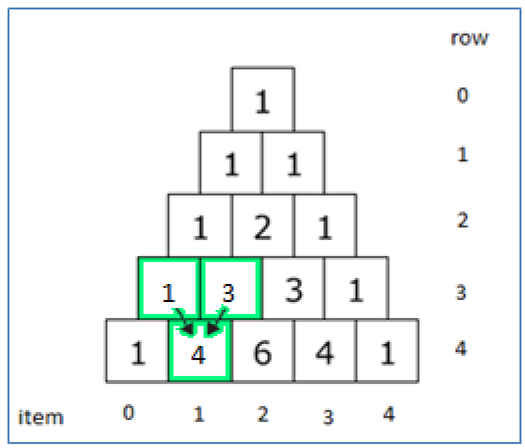
Sample Input	Sample Output
5 ACMICPC'14 12345678 abc12+ success$ 1Malaysia	Case #1: Acceptable Case #2: Not acceptable Case #3: Not acceptable Case #4: Acceptable Case #5: Acceptable

B	PASCAL TRIANGLE
Input	Standard Input
Output	Standard Output
Time Limit	15 seconds

Problem Description

Pascal triangle is one of the most interesting number pattern. It is named after Blaise Pascal, a famous French mathematician and philosopher.

To build the triangle, start with *1* at the top (row 0, item 0), then continue placing numbers below it in a triangular pattern. Each number is the sum of two numbers immediately above it (except for the edges, which are all *1*). As exemplified in the diagram, the **item value** for (row 4, item 1) is 3 + 1 = 4.



Given the row number ***n*** and item number ***k*** for (***n***, ***k***), you are to calculate the **item value**.

Input

First line of the input contains ***T*** the number of test cases (1 <= ***T*** <= 100). Each test cases consists of two integers ***n*** (0 <= ***n*** <= 50) and ***k*** (0 <= ***k*** <= ***n***).

Output

For each test case, the output contains a line in the format Case #x: ***M***, where x is the case number (starting from 1) and ***M*** is the **item value** for item (***n***, ***k***).

Sample Input Output

Sample Input	Sample Output
4 2 0 4 1 5 2 10 4	Case #1: 1 Case #2: 4 Case #3: 10 Case #4: 210

C	SULTAN'S HOLIDAY	
	Input	Standard Input
	Output	Standard Output
	Time Limit	3 seconds

Problem Description

As the personal secretary to the Sultan of Isketambola, you are to schedule the Sultan's holiday to among several competing activities. Suppose you have a set S = {*1, 2, . . . ,n*} of n proposed activities that could fit in the Sultan's holiday schedule of M slots, obviously the Sultan can only be doing one activity at a time. Each activity *i* has a start time s_i and a finish time f_i, where $s_i \leq f_i$. If selected, activity *i* takes place during the half-open time interval [s_i , f_i). Activities *i* and *j* are compatible if the intervals [s_i , f_i) and [s_j , f_j) do not overlap (i.e., *i* and *j* are compatible if $s_i \geq f_j$ or $s_j \geq f_i$). The idea is to select activities with a maximum-size set of mutually compatible activities.

Input

The first line of input gives the number of test cases, T ($1 \leq T \leq 10$), followed by a blank line. Each test case of input contains an integer M (1 <= M <= 5000), followed by pairs of "s_i f_i" (0<=| s_i |<=50000 , 1<=| f_i |<=50000, 0<= *i* <=100000), each on a separate line. Each test case of input is terminated by pair "0 0". Each test case will be separated by a single line.

Output

For each test case, your program should print the test case with a number (starting from one and incrementing at each test case) and the maximum number of activities which will be in the Sultan's schedule. In the following lines, the start times and final times, sorted by their start time(s_i) first and final times (f_i) second, should be printed in the same format as in the input. Pair "0 0" should not be printed. If no activities can fit in the [0,M] slots, then the maximum number of activities should be “0” (without quotes). Print a blank line between the outputs for two consecutive test cases.

Sample Input Output

Sample Input	Sample Output
3 4 0 1 2 3 4 5 0 0 14 1 4 3 5 0 6 5 7 3 8 5 9 6 10 8 11 8 12 2 13 12 14 0 0 2 0 3 1 4 0 0	Case #1: 2 0 1 2 3 Case #2: 4 1 4 5 7 8 11 12 14 Case #3: 0

D	ABC SERVICE MACHINE	
	Input	Standard Input
	Output	Standard Output
	Time Limit	10 seconds

Problem Description

ABC Service Machine provides a self-service for its customer. The customer must queue up in order to get the service. The customer will immediately be served if the queue is empty. If the machine is free, then it will serve the front customer in the queue. The machine takes 13 minutes to complete one service. The machine operation starts at 8.00 am and ends at 12.00 pm. However, it will continue serve customers who arrive before or by 12.00 pm.

The owner of the ABC Service Machine would like to know some statistical information so that he can improve the machine in the future. The information are:

- Number of customer that arrives by 12.00 pm.
- Longest customer waiting time.
- Average customer waiting time.

Assume that all customers are an ethical person and determined to get the service, also the machine is ideal.

Input

First line of input is an integer N (1 <= N <= 50), that represents the number of test case, followed by N lines where each line of input starts with an integer M (1 <= M <= 30) that represents the number of customer followed by M arrival time (in minutes) of the customers. The arrival time is a gap between arrivals except for the first customer, the time refer to a gap between starting time of the machine and his/her arrival.

Output

For each test case, the output contains a line in the format Case #x: A B C where x is the case number (starting from 1), A is an integer represent the number of customer that arrives by 12.00 pm, B is an integer represent the longest waiting time in minute and C is a real number with two decimal places represent an average customer waiting time.

Sample Input Output

Sample Input	Sample Output

3
22 14 10 15 12 11 14 11 12 13 10 13 10 14 10 13 10 10 10 14 13 13 10
16 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
2 0 10
	`Case #1: 20 23 10.15`
	`Case #2: 12 0 0.00`
	`Case #3: 2 3 1.50`

E	SPANNING TREE	
	Input	Standard Input
	Output	Standard Output
	Time Limit	5 seconds

Problem Description

Mrs. Balqis has just learned the concept of Minimum Spanning Tree. A spanning tree of a graph G is a subgraph of G that contains all vertices of G and form a tree (connected acyclic graph). Mrs. Balqis wonders if it is possible to find the minimum total weight spanning tree with a specific list of vertices as leaves (not necessarily the MST). However, Mrs. Balqis proves that such problem is NP-Hard. Feeling disappointed, Mrs. Balqis is now wondering another problem: find the minimum total weight spanning tree with a specific list of vertices as subset of the leaves. Please help Mrs. Balqis with her new problem!

Input

The first line of input consists of an integer T denotes the number of cases. It is followed by T blocks, each representing a case.

The first line of each case contains two integers: V ($2 \leq V \leq 50{,}000$) and E ($1 \leq E \leq 100{,}000$), the number of vertices and the number of edges.

The next E lines each consists of three integer: u, v, and w ($1 \leq u, v \leq V$ and $1 \leq w \leq 1{,}000$), describing an edge connecting vertices u and v with weight w. There is no two vertices connected by more than one edge at the same time and there is no edge connecting a vertex with itself. Furthermore, there are at least one path connecting any two different vertices.

The next line consists of a single integer L, the number of vertices in the list. The following line consists of L distinct numbers which describe the vertices which should be the leaves of the spanning tree.

Output

Output consists of T lines, each describes the solution for each case with the same order as in input.

Each case consists of a single line with the format "Case #i: S", where i represents the case number and S represents the minimum total weight of a spanning with a specific list of vertices as subset of the leaves. If there is no solution, S should be -1.

Sample Input Output

Sample Input	Sample Output
1 3 3 1 2 1 2 3 2 3 1 3 1 2	Case #1: 4

Explanation

The tree contains the edges: (1, 2), (1, 3), which is different from MST (the total weight of edges in the MST is 3).

F	SEQUENCE	
	Input	Standard Input
	Output	Standard Output
	Time Limit	5 seconds

Problem Description

Abdullah is playing with X sequences of integers, the length of the i-th sequence is L_i. The first sequence consists of (with order) 1, 2, ..., L_1, the second sequence consists of $L_1 + 1$, $L_1 + 2$, ..., $L_1 + L_2$, and so on.

Abdullah has another sequence S, which is initially empty. Repeatedly, he will do the following operations until each of the X sequences consists of no elements:

1. Choose one of the X sequences which contains at least one element
2. Take either the first or the last element of the sequence, remove it from the sequence, and append it at the end of sequence S.

Abdullah wonders how many possible configurations of S can be achieved. Help Abdullah by writing a program to do so!

Input

The first line of input consists of an integer T denotes the number of cases. It is followed by T blocks, each representing a case.

The first line of each cases contains an integer X (1 <= X <= 100,000). Then, it is followed by X lines. The i-th of the X lines contains a single integer L_i (1 <= L_i <= 10,000,000), the length of the i-th sequence that Abdullah plays with.

Output

Output consists of T lines, each describes the answer of a case in the same order with input. Each line is in the format “Case #i: C“, where i is the case number (starting from 1) and C is the number of possible configurations of S modulo 1,000,003.

Sample Input Output

Sample Input	Sample Output
1 2 2 2	Case #1: 24

Explanation

The sequences that Abdullah is playing with are:
Sequence 1: 1, 2
Sequence 2: 3, 4

Any permutations of 1, 2, 3, 4 is possible for S.

G	THE PERFECT CHOCOLATE CANDY	
	Input	Standard Input
	Output	Standard Output
	Time Limit	10 seconds

Problem Description

Chef Ismail is a fan of candies with chocolate and caramel, and has devoted much of his life to finding the perfect ratio of chocolate to caramel. He recently discovered the perfect ratio, but to his dismay, none of his favourite candy shops sell candies with exactly that ratio. So the chef crafted a plan to buy several candies, and melt them together into a larger candy so that the resulting candy will have the perfect ratio. How many candies will he have to buy? The chef may buy multiple candies of the same type.

Input

Input begins with an integer T ($1 \leq T \leq 5$) that represent the number of test cases. Each test case begins with N ($1 \leq N \leq 15$), the number of different candies available at the various shops followed by N+1 lines. The first N lines contains 2 integers $Chocolate_i$ ($1 \leq Chocolate_i \leq 500$) and $Caramel_i$ ($1 \leq Caramel_i \leq 500$) indicating the grams of chocolate and caramel respectively contains in the i^{th} candy. Line N+1 contains 2 integers desiredChocolate ($1 \leq desiredChocolate \leq 500$) and desiredCaramel ($1 \leq desiredCaramel \leq 500$), indicating the chef's desired ratio of chocolate to caramel.

Output

For each test case, print on a single line to represent the minimum number of candies the chef will have to buy. If it is impossible to produce a candy with the desired ratio, print -1 instead.

Sample Input Output

Sample Input	Sample Output
2 3 4 5 2 4 4 1 1 1 3 2 3 4 6 10 15 5 8	Case #1: 3 Case #2: -1

H	IS IT THE SAME?	
	Input	Standard Input
	Output	Standard Output
	Time Limit	2 second

Problem Description

Puan Ruqayyah, Madam Chong and Miss Lela teach their dyslexia students to identify characters. They tested randomly on strings of characters and numbers. They found that a few children get excited when they read certain series of strings. The teachers were curious and began to find out that these kids love strings that have series of characters that can be read similarly from right and from left.

There are not that many meaningful words and numbers with such pattern. The teachers are interested to collect these meaningful words and numbers to share with their dyslexia students. Examples of such strings are CIVIC, KAPAK, 2102012 (which is 2nd Oct 2012), 1102011 (which is 1st Oct 2011), among others.

Now, you are to help Puan Ruqayyah and her colleagues to select such strings. Write a program that reads in a sequence of characters and determine if it can be read similarly from right and from left.

Input

The first line will contain T, number of test cases. After the T line, it is followed by each with a string of alpha-numeric characters. The string will have less than 50 characters.

Output

Yes - if the input string can be read from right and left similarly, No – if it can't.

Sample Input Output

Sample Input	Sample Output
5 RACECAR MIRROR 1991 MALAM 20102010	Case #1: Yes Case #2: No Case #3: Yes Case #4: Yes Case #5: No

I	*MYSTICAL GOLDEN TIP ARROW*	
	Input	Standard Input
	Output	Standard Output
	Time Limit	10 seconds

Problem Description

The blood-thirsty Prince of Iskandariah wants to impress his neighboring countries by showing off his hunting skills. He intends to use a special mystical golden tip arrow to kill as many deers as possible with a single straight shot. As the personal royal guard to the Prince, he has ordered you to help him figure out, given a map of position of deers, how many can he kill with a single shot. To avoid the wrath of Prince Iskandariah, you rush to your work table and start writing a program to fulfill his request.

Input

The first line of input gives the number of cases, T ($1 \leq T \leq 10$). T test cases follow. Each one contains a number N, such that $100 \geq N \geq 1$, followed by N lines describing the positions of the deers. Each such line contains two numbers, with two decimal values, describing the Cartesian coordinates X and Y of a deer, such that $100.00 \geq X, Y \geq -100.00$. Positions may be repeated, but only count once.

Output

The output is comprised of one line for each test case. The line identifies the test case with a number (starting from one and incrementing at each test case), the number of deers and the maximum number of aligned deers. The exact format shall follow the sample output. If there is only one deer in the map then the maximum number of aligned deer is 1.

Sample Input Output

Sample Input	Sample output
3 5 0.00 0.00 0.00 0.00 1.00 1.00 1.00 0.00 0.00 1.00 2 0.00 0.00 0.00 0.00 6 0.00 2.00 0.00 0.00 1.00 1.00 1.00 0.00 0.00 1.00 0.00 -2.00	Case #1: 4 2 Case #2: 1 1 Case #3: 6 4

J	NUMBER SYSTEM	
	Input	Standard Input
	Output	Standard Output
	Time Limit	5 second

Problem Description

A numbering system consists of a number of symbols which represent each of its digits. For example, a base 2 numbering system (binary system) has only two symbols representing its digits --- 0 and 1. As for a base 3 numbering system, it has three symbols representing each of its digits --- 0, 1, and 2.

Each numbering system can only use those digits it has to represent all numbers. Hence, repeating and combining the symbols will produce a new number. For example, in a base 2 numbering system, after 1, the next number will be 10, followed by 11, and then 100. Similarly, in a base 3 numbering system, since there is only three possible symbols to be used, after 2, the numbers will be 10, 11, 12, 20, 21, 22, 100, 101, 102, 110 and so on.

Write a program which will list the first N numbers in the numbering system. For simplicity, limit the numbering system to bases from 2 to 10.

Input

First line of the input contains T (1 <= T <= 200), the number of test cases. For each test case, there will two numbers given. The first, B, is a number between 2 to 10 (inclusive), representing the base of a numbering system, while the second, N (1 <= N <= 50), is the first N numbers in the numbering system to be displayed.

Output

The output for each test case will be a single line containing the first N numbers in the numbering system.

Sample Input Output

Sample input	**Sample output**
3 3 4 8 10 5 16	Case #1: 0 1 2 10 Case #2: 0 1 2 3 4 5 6 7 10 11 Case #3: 0 1 2 3 4 10 11 12 13 14 20 21 22 23 24 30