CODEWARE'18

INTRA AUST TECH CONTEST

AUST CSE WEEK

INTRA AUST PROGRAMMING CONTEST SPRING-2018

Organized By	**Powered By**
AUST CSE SOCIETY	DEBUGGER 35

7th July, 2018

You get 17 pages

11 problems

&

240 minutes

Toph β

Problem Setters

Problem	Setter	Affiliation
A	Musfiqur Sanim	Software Engineer, Enosis Solutions Ltd.
B	Rajon Bardhan	Lecturer, Southeast University
C	Arnab Das	Software Engineer, Cefalo Bangladesh Ltd.
D	Ridowan Muhammad	Software Engineer, TigerIT Bangladesh Ltd.
E	Shakil Ahmed	Software Engineer, CodeMarshal
F	Sifat Shishir	Software Engineer, Enosis Solutions Ltd.
G	Mir Imtiaz Mostafiz Naved	Lecturer, Ahsanullah University of Science and Technology
H	Mahir Asef Kabir	Software Engineer, Enosis Solutions Ltd.
I	Shahariar Shibli	Student, Ahsanullah University of Science and Technology
J	Shagor Hasan	Software Engineer, Enosis Solutions Ltd.
K	Muhammad Ahasanuzzaman	Lecturer, Ahsanullah University of Science and Technology (Study Leave)

A	Juglu The Runners-Up Footballer	Time Limit: 1 sec

Juglu is one of the best footballer of AUST CSE. But he had never won the 'Winning Trophy' till now. As usual in yesterday's Final match, he was announced as the 'Man of the Tournament' but his team again got the 'Runners Up Trophy'. During the match, Juglu noticed that some of the players of his team were not giving their best. After some analysis he assume that they lost interest in Final because they didn't get their Jersey Number as they wanted.

To celebrate FIFA WORLD CUP'18, AUST is going to arrange an Intra Mini World Cup Tournament and Juglu wants to take the opportunity to win. But he doesn't want to face the same problem like before, so this time he wants to provide his teammates their favorite Jersey Number. He knows that each one of his teammates has a favorite number. Therefore, he wants to give his teammates the Jersey Number in such a way that the Jersey Number is a divisor of their favorite number.

If he has a squad of **n** teammates, each one of them should contain a distinct Jersey Number between **1** to **n**.

Input

Input starts with an integer **T (T <= 100)** denoting the number of test cases. Each case starts with a line containing an integer **n (3 <= n <= 20)** denoting size of squad. Next line contain **n** integer **ai(1 <= ai<= 10^18, 1 <= i <= n)** i-th player's favorite number.

Output

For each case, print the case number and if it is possible to fulfill the condition for all player then print **n** numbers where i-th number represent the jersey number of i-th player otherwise print "Again Runners-up". If there are multiple solutions then print the lexicographically smallest one.

Sample I/O

Input	Output
3 3 2 1 3 3 6 6 6 3 2 4 8	Case 1: 2 1 3 Case 2: 1 2 3 Case 3: Again Runners-up

B	Prime Summation	Time limit: 1 sec

This problem gives you a positive integer number which is less than or equal to **100000 (10^5)**. You have to find out the following things for the number:

1. Is the number prime number? If it is a prime number, then print **YES**.
2. If the number is not a prime number, then can we express the number as summation of unique prime numbers? If it is possible, then print **YES**. Here unique means, you can use any prime number only for one time.

If above two conditions fail for any integer number, then print **NO**. For more clarification please see the input, output section and their explanations.

Input

At first you are given an integer **T (T<=100)**, which is the number of test cases. For each case you will be given a positive integer **X** which is less than or equal **100000**.

Output

For every test case, print only **YES** or **NO**.

Sample I/O

Input	Output
3 7 6 10	YES NO YES

Explanation

Case – 1 Explanation: 7 is a prime number.

Case – 2 Explanation: 6 is not a prime number. 6 can be expressed as 6 = 3 + 3 or 6 = 2 + 2 + 2. But you can't use any prime number more than 1 time. Also there is no way to express 6 as two or three unique prime numbers summation.

Case – 3 Explanation: 10 is not prime number but 10 can be expressed as 10 = 3 + 7 or 10 = 2 + 3 + 5. In this two expressions, every prime number is used only for one time.

C	What a prediction!	Time Limit: 1 sec

In world cup season we see predictions from amazing talented animals to predict winners of certain matches. That's funny, right? However, this matter really bothers our **friend1** and **friend2**. To them it's an insult (Why should animals predict results of human game?). So they have found a way to have fun instead of watching these predictions.

They will try to predict how many goals will be in this tournament. Note that, they will only predict, whether it's possible or not is none of our concern. They will calculate this in a specific method. However, they have matches to watch. So, they have appointed you, an amazing programmer to solve the problem.

You are given two strings **S** and **P**. You have to get the result using these **P** and **S**. Let's say you search in **S** and find **P** starting from index **x**. Then you will have to delete the middle character from **P** and again search in **S** from the next position (**x+1**) with the new string $\mathbf{P_{new}}$. Finding the middle character is confusing right? Okay, let's say the length of **P** in a certain stage is **N**. If **N** is even, you will delete the $\mathbf{N/2}^{th}$ character. If **N** is odd, delete the $\mathbf{(N/2 + 1)}^{th}$ one. You will not have to perform delete operation when size of the string **P** becomes 1.

Say, we have a text **S** = *abacdabcd* and a string **P** = *abacd*. **P** is found in **S** and the starting position is 1. Delete the $\mathbf{3}^{rd}$ character *a* from **P**, so $\mathbf{P}_{new}$ will be *abcd*. We then start our search from position 2 in **S** and **P** is found in **S** at position 6. Now we delete the character $\mathbf{2}^{nd}$ 'b' from **P**, so $\mathbf{P}_{new}$ will be *acd*. Then we start to match from position 7 in **S** but no further matching is found. Therefore our goal prediction result is 2.

Input

First line contains a number **T**. **T** test cases follow. Each case contains two strings **S** and **P**.
Here,
1 ≤ **T** ≤ 5,
1 ≤ length of **S** ≤ 10^6
1 ≤ length of **P** ≤ 2 x 10^4
S, P will contain only lowercase alphabets.

Output

Print the result in a format of **Case n: m**. Here, **n** is the case number and **m** is the result.

Sample I/O

Input	Output
2 aaa a abcbabba abcba	Case 1: 3 Case 2: 2

D	Yet Another Longest Path Problem	Time Limit: 2 sec

We all know shortest path problem, right? When we have to move to one place from another using the shortest path. Now in this problem, we don't want shortest path but we are actually interested to find the longest path without exceeding some weight.

More specifically, you will be given a directed weighted graph with **N** nodes numbered **1** to **N**. Then you will be given a start node **U**, an end node **V** and a weight value **W**. You have to find the longest path from **U** to **V** where the cost of the path is maximum but not more than **W**.

Input

First line of the input will be one integer **N (1 ≤ N ≤ 100)**. Next **N** lines each will have **N** integers. The j^{th} integer of this i^{th} line, will be w_{ij} **(1 ≤ w_{ij} ≤ 100)**, weight of edge going to **j** from **i** or 0 if there is no road. w_{ii} will always be 0 that is will not be any self-loop.

In the next line of the input will be one integer **Q (1 ≤ Q ≤ 100)**, number of query. Next **Q** lines each will have three integers, **U V W (1 ≤ U, V ≤ N, 1 ≤ W ≤ 500)**.

Output

For each query, print the maximum cost of going **U** to **V** where the cost is not more than **W**. If it is not possible then print **-1**.

Sample I/O

Input	Output
5 0 2 4 0 0 3 0 0 0 0 9 2 0 2 0 0 0 0 0 1 0 6 0 0 0 5 3 1 5 3 1 6 3 1 11 1 3 39 1 3 3	5 5 10 39 -1

E	Strange Game	Time Limit: 1 sec

Shibli is very upset about early elimination of Germany from this world cup. So to cheer him up his friends decided to play a very strange game with him. There are **n** integer numbers are given **(a_1, a_2 …… a_n)**. Shibli needs to find an integer **x** so that the sum of the sequence **$(a_1 - x)^2 + (a_2 - x)^2 + \ldots + (a_n - x)^2$** is minimized. Can you please help Shibli to find the value of **x** and the minimum sum of the sequence? It is ensured that there is only one such **x** is present in this given sequence.

Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases. Each case contains an integer **n (1 ≤ n ≤ 10000)**, denoting how many numbers will be present for this test case. Next line contains **n** integer numbers say **x (1 <= x <= 10000).**

Output

For each test case, print a line **"Case x: y z"** where **x** is replaced by the test case number and **y** is the value of **x** described in the problem statement and **z** is minimum sum of the sequence.

Sample I/O

Input	Output
2 3 1 2 3 5 1 2 9 2 6	Case 1: 2 2 Case 2: 4 46

F	# Inside the range	**Time Limit: 1 sec**

You will be given two integers **X** & **Y**.
Then you will be given another integer **K**.
You have to tell whether **K** lies inside the range of **X** & **Y** [exclusive] or not.

Input

There will be **T (< 100)** number of test cases each having three integers **X**, **K** & **Y** per line.
-1000000 <= X, Y<= 1000000
-1000000 <= K <= 1000000

Output

For each case print "**case #z: Yeah!**" if **K** lies inside the range of **X** & **Y**.
Otherwise print "**case #z: Nah!!**" without the quotes, where **z** is the case number.

Sample I/O

Input	Output
2 1 2 3 10 20 30	case #1: Yeah! case #2: Yeah!

Note: Exclusive means just inside the range, not on the range.

G	# TikiTaka Snoozefest	**Time Limit: 1 sec**

Pavel Nedved, former prominent Juventus and Czech Republic Midfielder, decided to watch the second round match of Fifa World Cup Russia 2018 between former champion Spain and host Russia. He envisaged that it would be an exciting game. He was proved wrong. Russia stood most of the time near their penalty box. Spain, on the other hand, instead of attacking vigorously, passed the ball among themselves. Defenders of Spain, passed the ball among themselves; then they passed it to midfielders of their team. They passed among themselves too. Then they again passed the ball to defenders. Defenders, after passing the ball among themselves again for some times, passed it to attackers; and so on. Nedved, being so disgusted watching this snoozefest of football, started building a string which would summarize the buildup of the attacking move of Spain. He used only three characters for different positions: "**D**" for Defenders, "**M**" for Midfielders and "**A**" for attackers. How he build the string? Let's set up an example passing buildup.

1. At the beginning, Midfielders (M) passed the ball among themselves for 10 times. Then they passed it to one of the Defenders.
2. Defenders (D) passed the ball among themselves for 20 times. Then they passed it to one of the Attackers.
3. Attackers (A) passed the ball among themselves for 5 times. Then they passed it to one of the Midfielders.
4. Midfielders (M) passed the ball among themselves for 7 times. And somehow, the move was finished (the ball went out/ was intercepted etc.)

So, the string for this move is "MDAM", in order of the positions of passing build up. Total number of passes played in this move is: 10 + 1 + 20 + 1 + 5 + 1 + 7 = 45.

Nedved also heard that Spanish coach has asked the defenders to must play at least 10 passes in a single passing move among themselves (they played 20). He ordered the midfielders to play at least 6 passes in a single passing move among themselves (they played 10 and 7 in two different phases, both of which are greater than 6). For the Attackers, they were ordered to play at least 3 passes (they played 5). Nedved thought that under these constraints, this move could also be built in another way with same number of passes in total and in same order of positions of moves: 12 passes among Midfielders, 1 pass to Defenders, 13 passes among Defenders, 1 passes to Attackers, 8 passes among Attackers, 1 pass to Midfielders, 9 passes among Midfielders again finally; resulting in 12+1+13+1+8+1+9 = 45 passes. Nedved wondered that if such a move string **S** like "MDAM" is given with constraints like- at least **M** passes among Defenders, **N** passes among Midfielders and **Q** passes among Attackers have to be made (in the example, **M** = 10, **N** =6, **Q** =3); and it is said that exactly **P** passes are made; then in how many ways this passing move can be done in total?

Nedved knows someone almost having the same name as him who could help. The "almost-namesake" of Nedved has asked you to solve the problem.

Input

Input file will have an integer **T (T<=10000)** in first line, denoting the number of test cases. Then it will contain **3** lines for each of the test cases; resulting in **3T** separate lines. For each test case, the first line will denote the length **L (3 <= L <= 50)** of the string **S** denoting the passing move. **S** will be in second line of each test case. No two consecutive letters in **S** will be same. Then in the third line, 4 integers separated by a single space between themselves, **P, M, N, Q** are given. They are denoted exactly as stated previously. Here, **10<= P <= 1000**, **3 <= M, N, Q <= 10**.

Output

Print the result- in how many ways the given passing move under the constraints can be constructed- in lines, separated by a single newline after each test case. If you think no possible move exists, print **0**. As the results can be big, print them modulo **1000000007**.

Sample I/O

Input	Output
1 4 DMAM 8 1 1 1	4

Explanation:

In first sample, 4 possible passing build ups can be made under the constraints.

1. 2 Passes between Defenders, 1 pass to Midfielders, 1 pass between Midfielders, 1 pass to attackers, 1 pass between attackers, 1 pass to Midfielders, 1 pass between Midfielders.

2. 1 Pass between Defenders, 1 pass to Midfielders, 2 passes between Midfielders, 1 pass to attackers, 1 pass between attackers, 1 pass to Midfielders, 1 pass between Midfielders.

3. 1 Pass between Defenders, 1 pass to Midfielders, 1 pass between Midfielders, 1 pass to attackers, 2 passes between attackers, 1 pass to Midfielders, 1 pass between Midfielders.

4. 1 Pass between Defenders, 1 pass to Midfielders, 1 pass between Midfielders, 1 pass to attackers, 1 pass between attackers, 1 pass to Midfielders, 2 passes between Midfielders.

H	Not Argentina's match	Time Limit: 1 sec

There are **N (1 <= N <= 100)** spectators to watch a match between South Korea and Japan. Each spectator has an excitement level **E_i (1 <= E_i <= 100, 1 <= i <= n)** and a ticket worth of price **P_i (1 <= P_i <= 100, 1 <= i <= n)**. Now the stadium authority wants to allow spectators to enter the stadium but they don't want too many excitement in the stadium as there has been an ugly incident last match when some over excited spectators started beating opponent team's supporters. So now the authority is thinking of a way to allow spectators in such way that the ticket selling amount is maximum but the total excitement level is not over a number **W (1 <= W <= 1000)**.

Given **N**, **W** and **N** spectators' excitement **E** and ticket price **P**, write a program to find out the maximum ticket selling profit stadium authority can get without exceeding **W,** as total excitement level.

Input

First line of input is **T (1 <= T <= 100)**. Each of the next **T** lines will contain two integers **N**, **W** and each of the following **N** lines will contain two integers **E_i** and **P_i** where **i** is for the **i-th** spectator.

Output

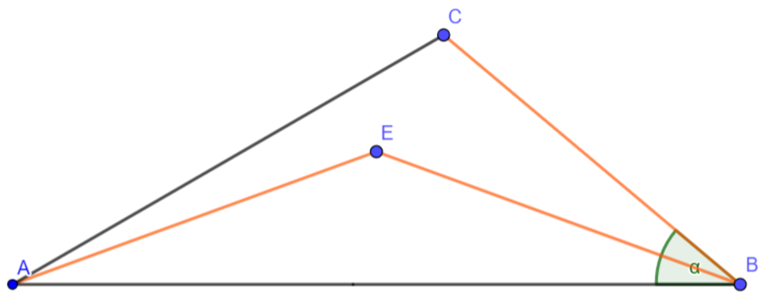
For each case, output the maximum profit authority can get.

Sample I/O

Input	Output
1 3 10 6 6 5 5 4 4	10

I	Angle Calculation	Time Limit: 1 sec

Techboy is watching the Quarter Final Match, Brazil vs. Belgium, of FIFA WORLD CUP 2018 on TV. At 90th minute, Neymar is being fouled. Brazil gets a free kick so they are trying to win the match by a "last-minute goal". Before taking the free kick Neymar, Coutinho, Willian is taking position. In our problem, Player **A** is Neymar, player **B** is Willian & player **C** is Coutinho. **E** is not any player, he is the referee! Techboy takes a pen and a paper, and draws the position of the players and referee as points and connects some of them. He gets the following figure.



Techboy knows that the distance between Neymar to Referee is **AE**, Willian to Referee is **BE** and Willian to Coutinho is **BC** which are same. The angle between Neymar-Willian-Coutinho which is $\angle \boldsymbol{ABC}$ and the angle between Referee-Neymar-Coutinho is $\angle \boldsymbol{EAC}$. **BE** divides $\angle \boldsymbol{ABC}$ into half. Given $\angle \boldsymbol{ABC}$, can you help Techboy to find $\angle \boldsymbol{EAC}$?

Input

The first line of the input is an integer **T (1 <= T <= 50)** denoting the test case. The next **T** lines contain a single integer **X (20 <= X <=70)** denoting the angle described in the problem.

Output

There will be **T** lines of output in the form of "**Case X: Y**", where **X** is the case number and **Y** is the value of $\angle \boldsymbol{EAC}$ which has two digits after the decimal point. See the sample I/O.

Sample I/O

Input	Output
2 40 50	Case 1: 10.00 Case 2: 8.22

J	Goal History	Time Limit: 1 sec

This problem is for the Brazil fans. You will be given the goal history of Brazil in world cups. The goal history will contain the years, when the world cups were held (not necessarily the actual world cup) and goal(s) scored by Brazil in these world cups. Then you will be given some queries, in each query you will be given a year when a world cup was held. You have to tell how many goal(s) was scored by Brazil in the world cup which came immediately before that world cup and in that world cup immediately after that world cup.

Input

First line of input will contain **T (1 <= T <= 10)**, the total number of test cases. Each of the test cases will contain **N (1 <= N <= 10^5)** and **Q (1 <= Q <= 10^4)** which will denote the total number of world cups and number of queries. Each of the next **N** lines will container two integers **Y (1 <= Y <= 10^18)** which will denote a year when a world cup was held and **G (1 <= G <= 20)** total number of goal(s) scored by Brazil in that world cup. Each of the next **Q** lines will container one integer which will denote a year when a world cup was held.

Output

For each of the **Q** queries of each test case you have to print total number of goal scored by Brazil in world cup held before the given year, in the given year and after the given year. If no world cup is held before or after the given year print **-1** instead. Print a blank line after each test case. See sample test case for clear understanding.

Sample I/O

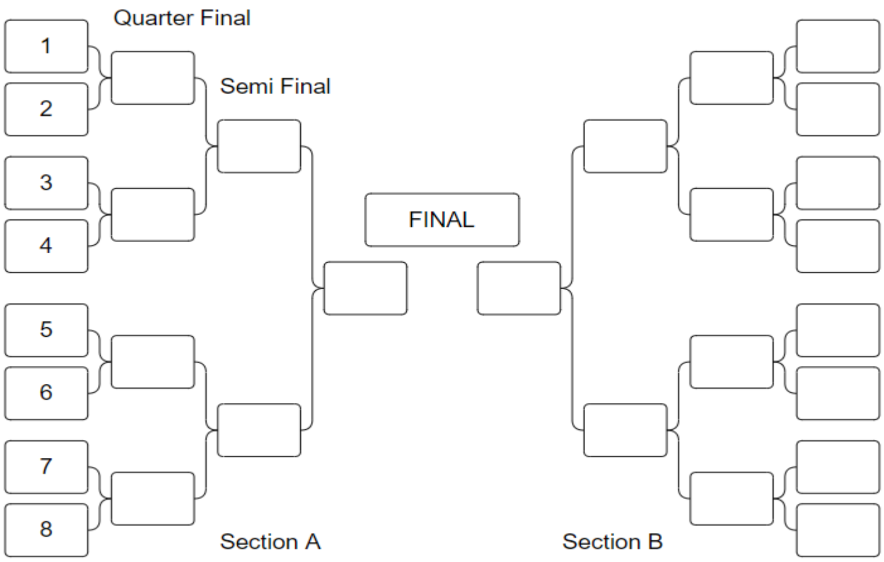
Input	Output
3 3 1 1998 1 2002 2 2006 3 2002 1 1 1998 2 1998 2 2 1998 2 2002 1 1998 2002	1 2 3 -1 2 -1 -1 2 1 2 1 -1
1	-1 1 2

3 3 1998 1 2002 2 2006 3 1998 2002 2006	1 2 3 2 3 -1

K	Final!!	Time Limit: **1 sec**

The knockout stage of the world cup has already begun and everyone is trembling with excitement for their favorite team in this world cup. Surprisingly, you are feeling relaxed because you already know whether your favorite team **(team 7)** has a good chance of going to the world cup final or not.

You have the data of the number of head to head matches between two teams and the result (win or lose). The figure shows the bracket of the knockout stage. The number in the box shows the position of the teams.



If you want to calculate the probability of going to the final of a team in Section A, you only require the data for the 8 teams in that section for your analysis. The explanation section describes one example process of calculating the probability.

NOTE: You will find the data at the end of the problem.

Input

The program does not require any input.

Output

Print the probability value (rounding up to two decimal places) of your favorite team going to the final.

Data and Explanation

See the next page for data & explanation.

Team (i)	Team (j)	Number of matches	Number of matches team i wins against team j	Number of matches team i loses against team j
1	2	20	5	15
1	3	30	15	15
1	4	5	2	3
2	3	8	3	5
2	4	6	6	0
3	4	10	5	5
5	6	12	8	4
5	7	10	6	4
5	8	6	4	2
6	7	9	4	5
6	8	1	1	0
7	8	3	2	1
1	5	4	0	4
1	6	7	4	3
1	7	6	2	4
1	8	20	12	8
2	5	8	4	4
2	6	4	2	2
2	7	6	2	4

2	8	11	10	1
3	5	5	4	1
3	6	6	4	2
3	7	5	3	2
3	8	6	1	5
4	5	5	3	2
4	6	2	1	1
4	7	8	3	5
4	8	7	2	5

We only consider the matches where a team wins or loses

Explanation:

For example, we want to calculate the probability of going to the world cup final for **team 1**.

Let, **W (i , j)** = the probability of team i wins against team j

L (i , j) = the probability team i loses against team j

Here,

W (1, 2) = 5/20 = 0.25

L (1, 2) = 1 – W (1, 2) = 0.75

W (2, 1) = L (1, 2) = 0.75

L (2, 1) = W (1, 2) = 0.25

(We do not need to consider the probability of the draw for a team)

First, we want to calculate the probability of the event of going to the semifinal of **team 1**. There are two possible ways that event can happen.

1. **Team 1** must wins against **team 2**, **team 3** can win against **team 4** and **team 1** must win against **team 3**
2. **Team 1** must win against **team 2** and **team 3** can lose against **team 4** (same as **team 4** can win against **team 3**) and **team 1** must win against **team 4**

Therefore, the probability of team 1 going to the semifinal,

WQF(1) = W(1,2) * W(3,4) * W(1,3) + W(1,2) * L(3,4) * W(1,4)

In this similar way, you can easily calculate the probability of going to the final of **team 1** and **the probability value is**

= WQF(1) * WQF(5) * W(1,5) + WQF(1) * WQF(6) * W(1,6) + WQF(1) * WQF(7)* W(1,7) + WQF(1) * WQF(8)* W(1,8)