HOSTED BY

DEPARTMENT OF CSE, BUET

Bangladesh Informatics Olympiad 2016

National Round

Problems A – F (8 pages including cover)

24-Dec-15

Problem A
Guess the Queue

Time Limit: 4 second

Busland is a city where the only means of transportation are buses. To get on a bus in Busland, people have to wait in queue. Each person has a **unique** bus **ID**. The person at the front is considered the first person and the person at the back is considered the last person in the queue. Trivially, the first person will get on the bus first and the last person will get on last.

It is well known that you should enter at the back of the queue. However, some people are very late and even though it is considered very rude, they try to enter in an alternative way. Since the sides of the queue are barricaded, the only alternative way is to enter from the front of the queue. Sometimes there are also people at the back of the queue, who become tired of waiting. They exit the queue from the back and just start walking to their destination instead.

Your task is to deal with three types of operations as follows:

1. '**1** ***x y***' The person with **ID** ***y*** enters from the back or front, if ***x*** is '**B**' or '**F**' respectively.
2. '**2** ***x***' The person in the back or front exits the queue, if ***x*** is '**B**' or '**F**' respectively.
3. '**3** ***x y***' Find the **ID** of the person in the y^{th} position of the queue or the position of the queue, where the person with **ID** ***y*** is currently situated, if ***x*** is '**D**' or '**P**' respectively.

Input:

The first line of input will contain the number of test cases, ***T*** **(1 ≤ *T* ≤ 5)**. Then ***T*** cases follow.

Each case starts with an integer ***N***, denoting the total number of operations. The following ***N*** lines will contain one of the three types described previously. The operations are in chronological order.

It is guaranteed that the given input is always valid. Thus, for the second type operation, the queue will always be non-empty and for the third type, the given position or **ID** will always exist in the queue. Also, a person who has already exited will not enter the queue again.

Constraints:

For easy version: **1 ≤ N ≤ 2000**. [25% of total score]
For hard version: **1 ≤ N ≤ 200000**. [100% of total score]
In general,
1 ≤ each ID ≤ 10^9, IDs are **unique** for each person
1 ≤ each position ≤ current size of the queue

Output:

For each case, print the case number on a single line, in the format "**Case x:**" where ***x*** is the case number. For each operation of the third type, output a single integer, which denotes the answer of the query.

Sample Input	Sample Output
1 7 1 B 1 1 B 2 1 F 3 3 D 3 2 B 1 F 4 3 D 2	Case 1: 2 3

Explanation of the Sample:

There is only 1 test case, in which we have to perform a total of 7 operations.
At first, the queue is empty. A person with **ID 1** and **2** enters from the back. After that, a person with **ID 3** enters from the front. Now there are **3** people in the queue and the **ID** of the **3rd** person in the queue is **2**.
The person in the back (who has **ID 2**) exits the queue. This leaves only two people remaining.
Finally, a person with **ID 4** enters from the front. Now there are again **3** people in the queue and the **ID** of the **2nd** person of the queue is **3**.

Problem B
Beautiful Factorial Game

Time Limit: 1 second

The statement of this problem is very simple. Given two number n and k, you need to find the maximum power of k (i.e. x) such that n! % k^x = 0. Here n! is the notation of n factorial. If you are not familiar with the notation,

n! = 1 * 2 * 3 * 4 * 5 * 6 ..... * n

Input:

First line of the input will contain an integer t (1 ≤ t ≤ 20) denoting the number of test case. The next t lines contain two integer number n and k as described above.

Constraints:

For easy version:	1 ≤ n ≤ 10, 2 ≤ k ≤ 10.	[20% of total score]
For harder version:	1 ≤ n ≤ 100000000, 2 ≤ k ≤ 100000000.	[100% of total score]

Output:

For each test case, print “Case t: x” where t is the test case number and x is the maximum power of k for which n! % k^x = 0.

Sample Input	Sample Output
2 5 2 1000 2	Case 1: 3 Case 2: 994

Explanation of the sample:

In the first test case, n = 5 and k = 2. So, n! = 120.
n! % 2^0 = 0
n! % 2^1 = 0
n! % 2^2 = 0
n! % 2^3 = 0
n! % 2^4 = 8
n! % 2^5 = 24
n! % 2^6 = 56
n! % 2^7 = 120

So, the answer should be 3.

Problem C
Counting Permutations

Time Limit: 1 second

In a planet far away from Earth, there is a beautiful country named Magic land. The children of this country play a lot of interesting games with numbers. One of the most popular games is called Inversion. In this game, you will be given numbers from 1 to N. They are given in a certain order. You need to calculate all the inversions in the given permutation of the numbers. S/he who can say it first correctly wins the game. An inversion occurs when there exists a pair of indices i and j such that i < j and given number at i^{th} position is greater than the number at j^{th} position.

For example, let us consider a permutation of numbers 1 to 5: 5, 1, 4, 2, 3. This permutation has the following inversions: (5, 1), (5, 4), (5, 2), (5, 3), (4, 2), (4, 3). Therefore, the number of inversion will be 6. The first person to tell this number correctly will win this game.

For this problem, we want to know how many permutations of the numbers 1, 2, .., N will have at least K inversions.

A permutation X is different from another permutation Y if there exists some *i (1≤ i≤ N)* for which the number in i^{th} position is different in these two permutations.

Input

The first line of input file contains the number of test cases, **T (1≤ T≤ 50)**. Then **T** cases follow:

Each case consists of one line which contains two integers: **N** and **K.**

Constraint

For easy version:	**1≤ N≤ 200** and **0≤ K≤ 300.**	[40% of total score]
For hard version:	**1≤ N≤ 2000** and **0≤ K≤ 3000.**	[100% of total score]

Output

For each case, print “**Case x: y**” in a separate line, where **x** is the case number and **y** is the number of permutations with at least **K** inversions. As the number can be very large, print **y modulo 10,007**.

Sample Input	Sample Output
3 3 1 2 1 3 2	Case 1: 5 Case 2: 1 Case 3: 3

Problem D
One Punch Man

Time Limit: 1.5 second

Saitama is the most powerful hero alive. Having apparently trained himself to superhuman conditions, Saitama faces a self-imposed existential crisis, as he is now too powerful to gain any thrill from his heroic deeds. Every monster he faces, is killed with only one of his punch.

There are N groups of monsters on a line, i^{th} of them is positioned at distance X_i from the beginning of that line and has V_i members. When Saitama punches the ground at point P, all monsters within R vicinity of that punch (meaning from P-R to P+R both inclusive) are killed by the shockwave. Saitama is tired after all his hero duties. So he will punch at most K times. Can you help him figure out, what is the maximum number of monsters he can kill?

Input:

The first line contains number of test case T. Then T test cases follow. First line of each test case contains three integers N, R ***and k***. Following N lines contains two space separated intergers X_i and V_i.

Constraints:

$T \le 10$, $1 \le N \le 10^5$, $1 \le V_i \le 10^4$, $1 \le K \le 50$.

For easy version:	$0 \le R, X_i \le 10^4$.	[40% of total score]
For hard version:	$0 \le R, X_i \le 10^8$.	[100% of total score]

Output:

For each test case, print a line "Case t: m" where t is the test case number and m is the the maximum number of monsters Saitama can kill.

Sample Input	Sample Output
2 4 3 1 6 10 12 110 19 100 24 30 5 3 2 3 3 5 2 3 8 10 5 0 5	Case 1: 130 Case 2: 23

Explanation of sample input:

In sample 1, Saitama can punch at position 21 or 22 to kill 130 monsters. If he punches at position 21 then the shockwave ranges from 18 to 24, killing monsters at 19 and 24. Similarly if he punches at 22 then shockwave will kill from 19 to 25. Punch at no other position will kill more monsters.
In Sample 2, Saitama can kill all monsters with 2 punches.

Problem E
Village Fair
Time Limit: 1 second

There are N houses in a village far away from here. They are numbered from 1 to N. For this problem let's assume, the houses each can accommodate infinite amount of people. From each house there is a directed path to exactly one other house. One of the houses is a grand house. There is a fair going on currently in the grand house. From this house there is no exit, and everybody can enjoy the festival here. Initially there is a little kid in each of the houses of the village. Each kid has some non-negative amount of joy (they are all pretty excited about the fair). It is guaranteed that there is exactly one simple way from each house to the grand house.

Every kid from their house starts their journey to the grand house. When they go from a house to some other house, their joy changes by some positive or negative amount. This change depends on the path.

Now, for each houses, you have to count the number of distinct joy values that will come to this house at some point of time. Note that if two or more persons with the same joy value will be counted only once. See the sample explanation for details.

Input:

You are given an integer N in the first line. The next line will contain N integers, the initial joy value of the kids in the houses. The next line also contains N integers, where the k^{th} integer is the id of the house where the kids can go to from the k^{th} house or 0 if it is the grand house. The fourth line will contain N integers, the amount of change of joy that will occur if one person goes through the path from starting at k^{th} house or 0 if it is the grand house.

Constraints:

For easy version:	1≤ N≤ 1000.	[30% of total score]
For hard version:	1≤ N≤ 100000.	[100% of total score]

Output:

Print N lines, each containing the number of distinct joy values that will come to this house at some point of time.

Sample Input	Sample Input
5 7 3 2 5 1 0 1 1 3 3 0 4 1 0 4	3 1 2 1 1

Explanation of Sample:

Initially each house will contain the kids with joy {7, 3, 2, 5, 1}. The first kid always stays in the grand house. The second kid will go to the grand house with joy 7. The 3rd kid will go to the grand house with joy 3. The fourth kid will go to 3^{rd} house with joy 5 and from there he will make it to grand house with 6 joys. Same goes for 5^{th} person. So the first house will see {7, 7, 3, 6, 6}. The second house will see {3}. The third house will see {2, 5, 5} the fourth house will see {5} and the fifth house will see {1}. So the number of distinct joy values of each house is {3, 1, 2, 1, 1}.

Problem F
Batman and Robin
Time Limit: 4 second

Batman and Robin like playing games.

They play games on days when Joker is sick and not causing any problems. Today is one of those days.

In the game, Robin asks Batman to guess the word he is thinking about. As Batman is the greatest detective in world, he almost read Robin's mind and guessed a word very close to what Robin was thinking.

Robin told Batman that, if they erase a letter, each from the word Robin is thinking about and the word Batman guessed, they will become same.

Now, before making the next guess, Batman wants to know the number of possible solutions.

Can you help Batman by calculating that? Note that, Batman is really smart, that's why Robin uses words from different languages to stop him solving it in no time.

Input

First line of input contains T, number of test cases. First line of each test case contains two integers, L, the length of Robin's word and A, the number of letters in the language Robin is using. Second line of input contains L integers, where i^{th} integer indicates order of the i^{th} letter in its alphabet. Since the size of the alphabet is huge, we are representing a letter with its order in the alphabet.

Constraints

$1 \le T \le 5$

For easy version:	$1 \le L \le 20$ (easy), $1 \le A \le 26$ (easy)	[30% of total score]
For hard version:	$1 \le L \le 60$ (hard), 1 $A \le 10^9$ (hard)	[100% of total score]

Output

For each test case, print the number of possible solutions of Robin's puzzle, **modulo** 10^9+7.

Sample Input	Sample Output
2 2 2 2 1 2 2 2 2	4 3

Explanation of Sample:

In the first case Robin's word length is 2 and size of the alphabet is also 2.
Robin's word is represented by "2 1". So the valid solutions are "1 1", "1 2", "2 1", "2 2".
In the second case Robin's word length is 2 and size of the alphabet is also 2.
Robin's word is represented by "2 2". So the valid solutions are "1 2", "2 1", "2 2".