Lab Mid-Term Exam (100)

Answer all of the following questions. Each of them contains 10 Marks

1   WAP that will take n integers into an array. Now find out the number of occurrences for each of the unique numbers. Each line of the output will be a unique number that exists in the array and its frequency. You can print them in any order.

Sample input	Sample output
8 2 8 1 3 2 6 4 3	1 => 1 2 => 2 3 => 2 4 => 1 6 => 1 8 => 1
4 2 2 10 2	2 => 3 10 => 1

2   WAP that will take n integers into an array A and m positive integers into array B. Now find the difference (set operation) of array A and B or (A-B).

Sample input	Sample output
8 7 8 1 5 2 6 4 3 6 1 3 6 0 9 2	7 8 5 4
3 1 2 3 2 4 5	1 2 3

3   Farina is a hard worker. But she also loves shopping a lot. Each day she earns some money and spends some on shopping. Fortunately, she has a credit card without any restrictions (it can have any amount of negative/positive balance). Initially, her credit card has a balance of 0. But she becomes **UPSET** when her credit card balance goes negative otherwise, she remains **HAPPY.**

Input:

The first line contains a number of Test Case T. Each of the test cases contains the following lines:
- The first line contains the number of days D when Farina has earned some money.
- The next line contains D space-separated integers indexed from 0 to D-1 which denotes the amount earned by Farina on an ith day.
- The next line contains D space-separated integers indexed from 0 to D-1 which denotes the amount spent by Farina for shopping on an ith day.
- The next line contains an integer Q that denotes the number of queries.
- Each of the following Q lines contains an integer x denoting the query of the status of Farina.

Output:

For each Test case there will be a new line where For each query Print 0 if Farina is **UPSET** otherwise print 1.

Sample	Input	Output	Explanation
1	2 5 10 5 2 3 1 8 15 1 2 10 3 0 1 3 3 7 10 1 3 2 7 1 2	1 0 0 1	Number of days: 5 Farina's earning: 10 5 2 3 1 Farina's expense: 8 15 1 2 10 Number of queries: 3 After 0th day Farina's balance is $(10 – 8) = 2$ which is positive, so she is **HAPPY**. After 1st day Farina's balance is $(10+5 - 8 - 15) = -8$ which is negative, so she is **UPSET.**
2	1 4 10 1 1 2 7 4 4 4 2 1 3	1 0	After 1st day Farina's balance is $(10+1 - 7 - 4) = 0$ which is not negative, so she is **HAPPY.**

4   WAP that will take (m x n) positive integer inputs into a matrix of dimension m x n. Now replace all the duplicate integers by -1 in that matrix. Finally display it.

Sample input	Sample output
3 3 1 7 3 7 4 5 3 5 6	1 7 3 -1 4 5 -1 -1 6
2 6 2 2 2 2 2 2 6 5 4 3 2 1	2 -1 -1 -1 -1 -1 6 5 4 3 -1 1

5   WAP that will take (n x n) integer inputs into a square matrix of dimension n (where n must be an odd number). Then calculate the sum of the integers based on the following position pattern (consider only the boxed position during the sum). Please see the input-output.

Sample input	Sample output
5 1 2 3 4 5 2 3 4 1 6 3 4 9 6 7 4 2 6 7 8 5 4 3 2 1	71
7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	25

6   Write down a program that will take n number of input from the users and create a Linear Linked List of n size. Replace all the even numbers in the list with -1 and display the List.

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

7   Given a singly linked list of size N. The task is to reverse every k node (where k is an input to the function) in the linked list. If the number of nodes is not a multiple of k then left-out nodes, in the end, should be considered as a group and must be reversed (See Sample 2 for clarification). Your task is to complete the function **reverseKNodes()** which should reverse the linked list **in a group of size k** and return the **head of the modified linked list**.

Sample	Input	Output	Explanation
1	8 1 2 2 4 5 6 7 8 4	4->2->2->1->8 ->7->6->5	The first 4 elements 1,2,2,4 are reversed first and then the next 4 elements 5,6,7,8. Hence, the resultant linked list is 4->2->2->1->8->7->6->5.
2	5 1 2 3 4 5 3	3->2->1->5->4	The first 3 elements are 1,2,3 are reversed first and then elements 4,5 are reversed. Hence, the resultant linked list is 3->2->1->5->4.

8   Given a linked list of size K and two integers M and N. Traverse the linked list such that you retain M nodes then delete the next N nodes, continue the same till the end of the linked list.

Sample	Input	Output
1	8 2 2 1 2 3 4 5 6 7 8	1->2->5->6
2	10 3 2 1 2 3 4 5 6 7 8 9 10	1->2->3->6->7->8

9　　　　Doubly linked list is one of the fundamental data structures. A doubly linked list is a sequence of elements, each containing information about the previous and the next elements of the list. In this problem, all lists have a linear structure i.e. each element except the first has exactly one previous element, and each element except the last has exactly one next element. **The list is not closed in a cycle.**

In this problem, you are given n memory cells forming one or more doubly linked lists. Each cell contains information about an element from some list. Memory cells are numbered from 1 to n.

For each cell i you are given two values:

l_i — cell containing the previous element for the element in the cell i;
r_i — cell containing the next element for the element in the cell i.

If cell i contains information about the element which has no previous element then $l_i = 0$. Similarly, if cell i contains information about the element which has no next element then $r_i = 0$.

Three lists are shown in the picture. For example, for the picture above the values of l and r are the following: $l_1 = 4$, $r_1 = 7$; $l_2 = 5$, $r_2 = 0$; $l_3 = 0$, $r_3 = 0$; $l_4 = 6$, $r_4 = 1$; $l_5 = 0$, $r_5 = 2$; $l_6 = 0$, $r_6 = 4$; $l_7 = 1$, $r_7 = 0$.

Your task is to unite all given lists in a single list, joining them with each other in any order. In particular, if the input data already contains a single list, then there is no need to perform any actions. Print the resulting list in the form of values l_i and r_i. **Any other action, other than joining the beginning of one list to the end of another, can not be performed.**

Input

- The first line contains a single integer n ($1 \le n \le 100$) — the number of memory cells where the doubly linked lists are located.

- Each of the following n lines contains two integers l_i, r_i ($0 \le l_i, r_i \le n$) — the cells of the previous and the next element of the list for cell i. Value $l_i = 0$ if an element in cell i has no previous element in its list. Value $r_i = 0$ if an element in cell i has no next element in its list.

- It is guaranteed that the input contains the correct description of a single or more doubly linked list. All lists have a linear structure: each element of the list except the first has exactly one previous element; each element of the list except the last has exactly one next element. Each memory cell contains information about one element from some list, each element of each list written in one of n given cells.

Output

Print n lines, the i-th line must contain two integers l_i and r_i — the cells of the previous and the next element of the list for cell i, after all, lists from the input are united in a single list. **If there are many solutions print any of them.**

Input	Output	Explanation
7 4 7 5 0 0 0 6 1 0 2 0 4 1 0	4 7 5 6 0 5 6 1 3 2 2 4 1 0	The Given Doubly Linked List is as follows: 6<-> 4<-> 1<-> 7 5 <-> 2 3 After the allowed Join Operation it can be as follows: 3 <-> 5 <-> 2 <-> 6<-> 4<-> 1<-> 7 Thus For 1, l_1 = 4, r_i = 7 For 2, l_1 = 5, r_i = 6 For 3, l_1 = 0, r_i = 5 For 4, l_1 = 6, r_i = 1 For 5, l_1 = 3, r_i = 2 For 6, l_1 = 2, r_i = 4 For 7, l_1 = 1, r_i = 0 *Note that the answer is also acceptable with 6<-> 4<-> 1<-> 7 <-> 5 <-> 2 <-> 3 4 7 5 3 2 0 6 1 7 2 0 4 1 5 is also acceptable.

10   You all know about the generic template we created in Module 23 as `MYSTACK.h`. Write a code that will provide the stack template with a new function named mid().

For Example: If **st** is a **Stack Object** and I call **st.mid()**, This will return the mid of element of the stack. Modify the Generic Stack Header file and Submit the Updated Template file as `YOURNAME_STACK.h`. (such as: `MEHEDI_STACK.h`)