Problem A

Given the heights of n towers and a value k. We need to either increase or decrease height of every tower by k (only once) where k > 0. The task is to minimize the difference between the heights of the maximum and the minimum tower after modifications, and output this difference.

For example, suppose, arr[] = {1, 15, 10} and k = 6 then, the maximum difference is 5. We change 1 to 7, 15 to 9 and 10 to 4. Maximum difference is 5 (between 4 and 9). We can't get a lower difference.

Input

The first line of input contains an integer T (1<=T<=100) denoting the number of test cases. Then T test cases follow. The first line of each test case contains a positive integer K (0 < K <=30). The second line of each test case contains a positive integer N (0 < N <= 30), denoting number of towers. The third line of the test cases contains N integers (0 <= A[i] <= 500) denoting the heights of N towers.

Output

For each test case, in new line print out the minimum difference of heights possible.

Sample Input/Output

|  |  |
| --- | --- |
| Input | Output |
| 3  2  4  1 5 8 10  3  5  3 9 12 16 20  4  6  100 150 200 250 300 400 | Case 1: 5  Case 2: 11  Case 3: 292 |
| 2  3  4  1 5 15 10  10  2  4 6 | Case 1: 8  Case 2: 2 |
| 3  3  2  6 10  6  6  1 10 14 14 14 15  2  3  1 2 3 | Case 1: 2  Case 2: 5  Case 3: 2 |

Problem B

Think of an array of numbers in which two types of operation that is rotation and deletion are performed. The process of doing these 2 operations is that first rotate the array in a clockwise direction then delete the last element. In short, rotate the array i th times and then deletes the i th last element. If the i th last element does not exists, then delete the first element present in the array. So your task is to find out which is the last element that you have to delete from the array so that the array becomes empty after removing it.

For example

A = {1,2,3,4,5,6}.

Rotate the array clockwise i.e. after rotation the array A = {6,1,2,3,4,5} and delete the last element that is {5} so A = {6,1,2,3,4}. Again rotate the array for the second time and deletes the second last element that is {2} so A = {4,6,1,3}, doing these steps when you reach 4th time, 4th last element does not exists so you have to delete 1st element i.e. {1} so A={3,6}. So continuing this procedure the last element in A is {3}, so output will be 3.

Input:

The first line of input contains an integer T (1<=T<=200) denoting the no of test cases. Then T test cases follow. Each test case contains two lines. The first line of each test case contains an integer N (1<=N<=100). Then in the next line are N space separated values of the array A (1<=A[i]<=10^7).

Output:

For each test case in a new line print the required result.

Sample Input/Output

|  |  |
| --- | --- |
| Input | Output |
| 2  4  1 2 3 4  6  1 2 3 4 5 6 | Case 1: 2  Case 2: 3 |
| 2  3  1 3 5  2  1 2 | Case 1: 5  Case 2: 2 |
| 3  5  1 4 10 3 5  10  1 2 3 4 5 6 2 8 9 12  4  1 10 14 18 | Case 1: 10  Case 2: 4  Case 3: 10 |

Problem C

There are tuples given for each users of a website (Si,Ei) where Si denotes the when the user entered the website and Ei denotes when the user exits the website .Find the maximum number of users active of website at any time duration.

Example: Sil[] = {1, 2, 9, 5, 5}

Ei[] = {4, 5, 12, 9, 12}

First user in array arrives at 1 and leaves at 4, second user arrives at 2 and leaves at 5, and so on. Output is 5. There are maximum 3 users at time 5.

Input:

The first line of input contains an integer T (1<=T<=200) denoting the number of test cases. Then T test cases follow. Each test case contains three lines. The first line of each test case contains an integer N (1<=N<=100). Then in the next two lines are N space separated values of the array Si (1<=S[i]<=24) and Ei (1<=E[i]<=24).

Output:

For each test case in a new line print the maximum number of users active in the website at any time duration.

Sample Input/Output

|  |  |
| --- | --- |
| Input | Output |
| 2  5  1 2 10 5 5  4 5 12 9 12  3  1 2 10  4 5 12 | Case 1: 3  Case 2: 2 |
| 2  4  1 2 10 5  5 12 11 12  6  1 2 3 4 5 6  2 3 4 5 6 7 | Case 1: 3  Case 2: 2 |
| 2  2  1 3  2 5  3  3 3 3  4 5 6 | Case 1: 1  Case 2: 3 |

Problem D

Given an array of random numbers. Find longest increasing subsequence (LIS) in the array.

Input:

The first line of input contains an integer T (1<=T<=200) denoting the number of test cases. Then T test cases follow. Each test case contains two lines. The first line of each test case contains an integer N (1<=N<=100). Then in the next line are N space separated values of the array Ai (1<=A[i]<=10^7).

Output:

For each test case in a new line print the length of the LIS.

Sample Input/Output

|  |  |
| --- | --- |
| Input | Output |
| 4  8  10 22 9 33 21 50 41 60  4  3 4 5 10  4  10 5 4 3  10  9 2 5 3 7 11 8 10 13 6 | Case 1: 5  Case 2: 4  Case 3: 1  Case 4: 6 |

Problem E

Given an array of n positive integers. Write a program to find the sum of maximum sum subsequence of the given array such that the integers in the subsequence are sorted in increasing order. For example, if input is {1, 101, 2, 3, 100, 4, 5}, then output should be 1, 2 , 3 , 100 whose sum is 106, if the input array is 3, 4, 5, 10, then output should be 3 , 4 , 5 , 10 and sum is 22 and if the input array is 10, 5, 4, 3 then output should be 10. You have to print the sum.

Input:

The first line of input contains an integer T (1<=T<=200) denoting the number of test cases. Then T test cases follow. Each test case contains two lines. The first line of each test case contains an integer N (1<=N<=100). Then in the next line are N space separated values of the array Ai (1<=A[i]<=10^7).

Output:

For each test case in a new line print the maximum sum.

Sample Input/Output

|  |  |
| --- | --- |
| Input | Output |
| 4  7  1 101 2 3 100 4 5  4  3 4 5 10  4  10 5 4 3  10  9 2 5 3 7 11 8 10 13 6 | Case 1: 106  Case 2: 22  Case 3: 10  Case 4: 45 |