DAA Lab Qs

Week 0:

Question-1: Given an array of nonnegative integers, design a linear algorithm and implement it using a program to find whether a given key element is present in the array or not. Also, find the total number of comparisons for each input case. (Time Complexity = $O(n)$, where n is the size of input).

For this problem you are free to choose programming language, no. of test cases, input and output format.

If you feel ambitious about problem solving, keep the input inside a text file, read that using a program to generate the output in another file.

Hint - You can take the first line as n, no. of test cases and after that n lines containing various numbers of elements where each line represents an array.

Week 1:

Question-2: Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find the total number of comparisons for each input case. (Time Complexity = $O(logn)$, where n is the size of input).

Question-3: Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether a given key element is present in the sorted array or not. For an array $arr[n]$, search at the indexes $arr[0], arr[2], arr[4], .., arr[2^k]$ and so on. Once the interval ($arr[2^k] < key < arr[2^{k+1}]$) is found, perform a linear search operation from the index 2^k to find the element key. (Complexity < $O(logn)$, where n is the number of elements need to be scanned for searching): **Jump Search**

Input format:

For each test case, there will be three input lines.

The first line contains number of test cases, T.

First line contains n (the size of the array).

Second line contains n space separated integers describing the array.

Third line contains the key element that need to be searched in the array.

Output format:

The output will have T number of lines.

For each test case, output will be "**Present**" if the key element is found in the array, otherwise "**Not Present**".

Also for each test case output the number of comparisons required to reach the key.