CS5103/MC5101: Assignment 5 **Date: 28/08/24**

Question 1:

Implement a linear search function to find an element in a given LinkedList. If an element is present, return 1 else, return 0.

2M

```
struct node

{

        int data;

        struct node *next;

};

int linear_search(struct Node* head, int key)

{

        //write your code here

}
```

Question 2:

Rohan is a lover of sweets. Today, he went over to have some of it. To his surprise, the waiter turns out to be an Ex-SDE from amazon as he was laid off and refuses to serve him unless Rohan solves his problem, stated as:

Given two non-increasing array of integers A,B i.e **A[i] >= A[i+1]** and **B[i] >= B[i+1]** and

for all i, 0 ≤ i < n-1.

The monkiness of two numbers is given by: **M (A[i],B[j]) = j - i , if j >=i and B[j] >= A[i], or 0 otherwise.**

Find the monkiness of the two arrays, that is given by: M (A,B)= max (M(A[i],B[j])) for 0≤ i, j< n-1.

15M

Input Format:

The first line contains an integer, tc, denoting the number of test cases. The next line contains an integer, n, denoting the size of the two arrays. The size of both the arrays will be equal. After that line, the next line contains n integers denoting the numbers in the array A, and in the next line, there will be n numbers denoting the numbers in the array B.

Output:

Print the monkiness of the two arrays.

```
Example:                                Output:

2                                       5

9

7 7 3 3 3 2 2 2 1

8 8 7 7 5 5 4 3 2

6                                       0

6 5 4 4 4 4

2 2 2 2 2 2
```

Note: **Your approach should be optimal. The evaluation also depends on your code optimization.**

Question 3:

Implement the Ternary Search algorithm in an iterative method. Given a sorted array and a target value, return the index of the target value using ternary search. **If the target is not found, return -1.   3M**