1BM19CS085
Md Ibaduddin Saffan

# ADA LAB Test - 1

NAME: Md Ibaduddin Saffan

USN: 1BM19CS085

Course-code: 19CS4PCADA

Date : 08-06-21

2) Recursive Binary Search and Linear Search. Determine the time required to search an element. Repeat the experiment for different values of N and plot a graph of the time taken vs N.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void createArray (int a[], int n)
{
        for (int i=0; i<n; i++)
            a[i] = rand() % 100;

}
void display (int a[], int n)
{
        for (int i=0; i<n; i++)
            printf("%d   ", a[i]);

        printf("\n");
}
int LinearSearch (int a[], int start, int end, int key)
{
        if (start <= end)
        {
            if (a[start] == key)
                return start;
            return LinearSearch (a, start+1, end, key);
        }
        return -1;
}
```

```
int  Binary Search (int a[], int start , int end , int key )
{
       if ( start <= end )
       {
           mid  = start + (end - start)/2 ;
           if (a[mid] == key) return mid ;
           else if ( a[mid] < key ) ~~return~~
                return  BinrySearch ( a , mid+1, end , key);
           else
              return  Binry Search ( a; start, mid-1 , key);
       }
       return -1;
}
void   sort (int a[] , int);
void   main ()
{
     int  a [10000], n, key ~~;~~ c=1, choice ~~0~~, pos ;
     clock_t start, end;
     ~~prin~~
     while (c==1)
     {
        printf (" Enter size of array : \n");
        scanf (" %d", &n);
        create Array (a, n);
        printf (" Enter key to be searched : \n");
        scanf (" %d", & key);
        printf(" Linear Search : \n");
        start = clock ();
        pos  = Linear Search (a , 0, n-1, key);
        ~~if~~ end = clock ();
        if (pos == -1) printf("Not Found \n");
        else
           printf ("Found at position : %d\n", pos+1);
        printf(" Time taken : %f ms", (double)(end-start)/(CLK_TCK * 100);
```

ADA - LAB Test 1

1BM19CS085
Md Ibaduddin Saffan

```
        printf(" Binary Search :\n");
        sort (a,n);
        start = clock();
        pos = BinarySearch (a, 0, n-1, key);
        end = clock();
        if (pos == -1) printf("Not Found \n");
        else
            printf(" Found at position : %d \n", pos+1);
        printf(" Time : %f ms", (double) (end-start)/CLK_TCK *100);
        printf(" Press 1 to exist \n Press 2 to Continue \n");
        scanf(" %d", & choice);
        if (choice == 1) break;
    }
}
void    sort (int a[], int n)
{   int temp;
    for (int i=0; i<n; i++)
    {
        for (int j=i; j<n-1-i; j++)
        {
            if (a[j] > a[j+1])
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
}
```

Modification :

Find mode of key element :

```
int   count = 0 ;
int   mode ( int  a[] ,int start ,int end ; int key )
{
        if ( start <= end )
        {   mid = start + (end - start)/2 ;
            if (a [mid] == key), count ++ ;
            else if (a [mid] < key)
                    return BinarySearch (a, mid+1 , end ,key) ;
            else
                  return Binary Search (a, start, mid -1 , key ) ;
        }
        return count ;
}
```