15/4/21

ADA-LAB (WRITE-UP)

1) Recursive Linear and Binary Search.

```
#include <stdio.h>
int RecursiveLS (int arr[], int value, int index, int n)
 {
  int pos=0;
  if (index>=n)
   {
     return 0;
   }
  else if (arr[index] ==value)
   {
     pos = index+1;
     return pos;
   }
  else
  {
    return RecursiveLS (arr, value, index+1, n);
  }
  return pos;
}
int main()
{
 int n, value, pos, m=0, arr [100];
printf("Enter the total elements in the array:");
scanf("%d", &n);
printf("Enter the array elements: \n");
for(int i=0; i<n; i++)
 {
   scanf ("%d", &arr[i]);
 }
```

```
printf("Enter the element to search:");
scanf("%d", &value);
 pos = RecursiveLS(arr, value, 0, n);
  if (pos != 0)
 {
  printf("Element found at pos %d\n", pos);
  }
 else
 {
  printf("Element not found\n");
 }
 return 0;
}


# include<stdio.h>
void binary_search(int[], int, int, int);
void bubble_sort(int[], int);
 int main()
 {
  int key, size, i;
  int list[25];
 printf("Enter size of a list:");
 scanf("%d", &size);
printf("Enter elements\n");
 for(i=0; i<size; i++)
  {
    scanf("%d", &list[i]);
  }
 bubble-sort(list, size);
 printf("\n");
printf("Enter key to search\n");
```

```
scanf ("%d", &key);
binary-search (list, 0, size, key);
 }
void bubble-sort (int list [], int size)
 {
    int temp, i, j;
   for ( i=0; i < size; i++)
    {
       for (j=0; j < size ; j++)
        {
           if ( list [i] > list [j])
             {
                temp = list [i];
                List [i] = List [j];
                list [j] = temp;
             }
        }
    }
 }
void binary-search (int list[], int lo, int hi, intkey)
 {
    int mid;
    if (lo>hi)
   {
    printf ("Key not found \n");
    return;
   }
  mid = (lo+hi)/2 ;
  if ( list [mid] == key)
 {
   printf ("Key found \n");
 }
```
Scanned with CamScanner

```
else if( list[mid] > key)
{
binary-search (list, lo, mid-1, key);
}
else if (list[mid] < key)
 {
binary-search (list, mid+1, hi, key);
 }
}
```

e) Recursive GCD and Iterative GCD.

```
# include <stdio.h>
int gcd( int m, int n)
 {
   if (n==0) return m;
   if (m<n) return gcd (n,m);
   return gcd (n, m%n);
 }
  int main ()
 {
    int m,n, res;
  printf ("Enter m and n \n");
  scanf ( "%d %d", &m, &n);
  res = gcd (m,n);
  printf ("GCD of %d and %d is %d. \n", m, n, res );
 }
```

```
#include <stdio.h>
 int gcd (int m, int n)
 {
     int r;
     do
   {
      r = m%n;
      m = n;
      n = r;
   } while (n!=0) ;
    return m;
   }
   int main()
 {
   int  m, n, res;
   printf ("Enter m and n\n");
   scanf  ("%d %d", &m, &n);
   res= gcd (m,n);
   printf ("The GCD of %d and %d is %d. \n", m, n, res);
   }
```