Linear search (Recursive) 15/4/21

LAB-1

```
#include <stdio.h>
int recsearch (int arr[], int l, int r, int x)
{
if (x < l)
 return -1;
if (arr[l] == x)
    return l;
if (arr[r] == x)
    return r;
return recsearch (arr, l+1, r-1, x);
}
int main ()
{
int x;
int arr[] = {12, 34, 54, 2, 3}, ;;
int n = sizeof(arr) / sizeof(arr[0]);
print ("enter the element\n");
Scanf ("%d", &x);
int index= recSearch (arr, 0, n-1, x);
 if (index != 1)
 printf ("element %d is present at index", x,
  else
  printf (" element %d is not present", x);
 return 0;
}
```

BINARY SEARCH

```
#include <stdio.h>
int bs(int a[], int l, int r, int x)
{
int mid = (l+r)/2;
 if (l > r) return -1;
if (a[mid] == x) return mid;

if (a[mid] < x)
  return bs(a, mid+1, r, x)
  else
  return bs(a, l, mid-1, x)
}
int main(void)
{
int a[10];
int len, pos, seach_item;
printf("enta length of array\n")
scanf("%d", &len);
printf("enta array elements\n")
for(int i=0; i<len; i++)
scanf("%d", &a[i]);
printf("enta the element to search\n")
 scanf("%d", &seach_item);
pos = bs(a, 0, len-1, seach_item);
if (pos<0)
printf("can't find\n");
else
printf("position of %d in the array is %d\n", seach
return 0;
}
```

GCD - Recursive

LAB-1
15/4/21

```
#include<stdio.h>
int gcd(int a, int b)
{
if (a==0) return b;
  return gcd(b%a, a)
}
int main()
{
int a,b;
printf("enter value of a,b\n");
scanf("%d %d", &a, &b);
 printf("GCD(%d,%d)= %d", a,b, gcd(a,b);
 return 0;
}
```