

## \* LAB - 1 \* PROGRAMME - 1 \*

### \* GCD RECURSIVE :

```
#include <stdio.h>
int gcd (int n1, int n2)
{
    int rem;
    if (n2 == 0) return n1;
    else
        return gcd (n2, n1 % n2);
}
```

```
int main () {
    int num1, num2, gcd;
    printf ("Enter two numbers");
    scanf ("%d %d", &num1, &num2);
    gcd = gcd (num1, num2);
```

```
    printf ("gcd is %d", gcd);
```

```
}
```

### \* GCD ITERATIVE :

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

```
int main () {
```

```
    int num1, num2, rem;
```

```
    printf ("Enter two numbers");
```

```
    scanf ("%d %d", &num1, &num2);
```

```
    if (num1 < num2) {
```

```
        num1 = num1 + num2;
```

// swap

```
        num2 = num1 - num2;
```

```
        num1 = num1 - num2;
```

```
}
```

while (num2 != 0)

{

rem = num1 % num2;

num1 = num2;

num2 = rem;

}

printf("gcd is %d", num1);

}

\* PROGRAMM - 3 \*

BINARY RECURSIVE

#include <stdio.h>

int binaarysearch (int arr[], int key, int n1, int n2) {

int mid = n2/2;

if (arr[mid] == key) {

printf("key id found %d", mid);

if (arr[mid] < key)

return binaarysearch (arr, key, n1, mid);

if (arr[mid] > key)

return binaarysearch (arr, key, mid, n2);

}

// main binaarysearch function.



```
void swap (int *xp, int *yp)
```

```
{
```

```
    int temp = *xp;
```

```
    *xp = *yp;
```

```
    *yp = temp;
```

```
}
```

// swapping elements

```
void selectionSort (int arr[], int n) {
```

```
    int i, j, min_idx;
```

// sorting the array

```
    for (i = 0; i < n - 1; i++) {
```

```
        min_idx = i;
```

```
        for (j = i + 1; j < n; j++)
```

```
            if (arr[j] < arr[min_idx])
```

```
                min_idx = j;
```

```
        swap (&arr[min_idx], &arr[i]); // interchange values
```

```
}
```

```
int main() {
```

```
    int i, n, key;
```

```
    pf ("Enter size");
```

```
    scanf ("%d", &n);
```

```
    int arr[n];
```

```
    pf ("Enter elements");
```

```
    for (i = 0; i < n; i++)
```

```
        { scanf ("%d", &arr[i]);
```

```
    }
```

```
    selectionSort (arr, n);
```

```
    pf ("Enter the key");
```

```
    sf ("%d", &key);
```

```
    binarySearch (arr, key,
```

```
0, n);
```

```
}
```

# \* LINEAR SEARCH PROGRAM \*

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

```
int main() {
```

```
    int arr[10], i, n, key, flag = 0;
```

```
    printf("enter no. elements in array");
```

```
    scanf("%d", &n);
```

```
    printf("enter items into array");
```

```
    for(i=0; i<n; i++)
```

```
    {
        scanf("%d", &arr[i]);
    }
```

```
    printf("enter the elements need to be founded: \n");
```

```
    scanf("%d", &key);
```

```
    for(i=0; i<n; i++) {
```

```
        if(arr[i] == key) {
```

```
            printf("key found in position %d", i);
```

```
            flag = 1;
```

```
        }
```

```
    }
```

```
    if(flag == 0) {
```

```
        printf("key not found");
```

```
    }
```