AIM: IMPLEMENTATION OF BINARY SEARCH

a) Iterative Method:

```
#include<stdio.h>
int binarysearch(int array[], intx, int low, int high){
  while(low<=high){
    int mid = low + (high-low)/2;
    if(array[mid]==x)
      return mid;
    if(array[mid]<x)
      low = mid+1;
    else
      high= mid-1;
  }
  return -1;
}
int main(void){
  int array[]={3,4,5,6,7,8,9};
  int n= sizeof(array)/size_of(array[0]);
  int x=4;
  int result=binarySearch(array,x,0,n-1);
  if(result==-1)
    printf("Not found");
  else
    printf("Element at index %d", result);
    return 0;
}
```

NAME

DATE

UNIV.ROLL

ALGORITHM:

do until the pointers low and high meet each other

```
mid = (low + high)/2
If (x == arr[mid])
    return mid
else IF (x > arr[mid])
    low = mid + 1
else
    high = mid - 1
```

b) Recursive Method

```
#include<stdio.h>
int binarysearch(int array[],int x, int low, int high) {
if(high>=low) {
  int mid = low+(high-low)/2;
  if(array[mid]==x)
    return mid;
  if(array[mid]>x)
    return binarysearch(array,x,low,mid-1);
  return binarysearch(array, x, mid+1, high);
}
return -1;
}
int void(main) {
    int array[] ={3,4,5,6,7,8,9};
    int n= size of(array)/sizeof(array[0]);
    int x=4;
    int result = binarysearch(array,x,0,n-1);
    if(result==-1)
      Printf("Not found");
    else Printf("Element is found at index %d", result);

}
```

ALGORITHM :

Linear_search (a, n, val)

'val' is the value to search

step 1: set Pos = -1

step 2 : set i = 1

step 3 : repeat step 4 while i<=n

step 4: if a[i] == val

set Pos = i

Print Pos

go to step 6

[end of IF]

set i = i+1

[end of loop]

step 5: IF POS = -1

Print "value is not Present in the array"

[end of IF]

step 6: exit

SEMESTER

# GIET UNIVERSITY,GUNUPUR

SHEET NO.

AIM: IMPLEMENTATION OF LINEAR SEARCH

Program:

```
#include<stdio.h>
int search(int arr[], int n, int x)
{
  int i;
  for(i=0;i<n;i++)
      if(arr[i]==x)
        return i;
    return -1;
}
int main(void)
{ int arr[]={ 2,3,4,10,40};
  int x=10;
  int n= sizeof(arr) / sizeof(arr[0]);
  int result = search(arr,n,x);
  (result == -1)
     ?printf("Element is not Present in array ")
     :printf("Element is at index %d", result);
  return 0;
}
```