Experiment No. 2

Aim: Implement Binary Search using Divide & Conquer Approach

Theoretical Background:

A binary search algorithm is a technique for finding a position of a specified value within a sorted array.

- Algorithm:

```
Low=1
High=n
While(low<=high)
{
    Mid=(low+high)/2
    If(a[mid]==x)
        Return mid
    Else if(x<a[mid])
        High=md-1
    Else
        Low=mid+1
}
Return-1
}
```

Program:

```
#include <stdio.h>
#include<conio.h>
    int binarySearch(int array[], int x, int low, int high)
{
   // Repeat until the pointers low and high meet each other
    int n;
    low=array[0];
```

```
        high=array[n];
        while (low <= high)
            {
                int mid = low +high / 2;
                 if (array[mid] == x)
                        return mid;
                if (array[mid] < x)
                        high = mid-1
                else
                        low = mid +1;
          }
          return -1;
           }
int main(void)
 {
  int array [30];
  int x,result,n,i;
  clrscr();
  printf("\n Enter the size of array:");
  scanf("%d",&n);
  printf("\nEnter array elements:");
  for(i=0;i<n;i++)
  {
  scanf("%d",&array[i]);
  }
```

```
  printf("\nEnter the element to be searched:") ;
  scanf("%d",&x);
  result = binarySearch(array, x, 0, n);
  if (result == -1)
    printf("Not found");
  else
    printf("Element is found at index %d", result);
  getch();
  return 0;
}
```

Output:

Conclusion:

Thus, in this experiment we have studied about Binary Search and how to solve it by using Divide and Conquer Approach.