Name	**PRATIK PUJARI**
UID no.	**2020300054**
Experiment No.	**4**

AIM:	
Program 1	
PROBLEM STATEMENT :	The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from the unsorted part and putting it at the beginning. The algorithm maintains two sub-arrays in a given array.
PROGRAM:	ALGORITHM: void main(): STEP 1: START. STEP 2: Initialize the variable n , loop counter(i) and array “a”. STEP 3: Input the size of the array and store it in the variable name “n”. STEP 4: Printf(“Enter the elements of the array “) STEP 5: For i=0 and less than n , Repeat the step 5.1 and 5.2 else if the condition fails, go to step 6. STEP 5.1: Take the input of the user and store it in a[i]. STEP 5.2: Increment the loop counter by one . STEP 6: printf("\n The sorted array is :"). STEP 7: Call the predefined function sel_sort(a,n) STEP 8: Initialize a variable under the name “num”. STEP 9: Input the number to be searched by binary search and store it in the variable name “num”. STEP 10: If bin_search(a,num,n)is equal to zero then printf “Number not found”else , go to Step 11. STEP 11: printf("The number is at %d position", bin_search(a,num,n)). STEP 12: END. sel_sort(int a[ ],int n) STEP 1: START. STEP 2: Initialize the variable “temp” , i=0 and j. STEP 3: For i equal to zero and less than n, Repeat the steps 3.1 and 3.2 or else if the condition fails go to step 4. STEP 3.1: For j equal to i and less than n ,Repeat the steps 3.1.1 and 3.1.2 or else if the condtion fails go to Step 3.2 STEP 3.1.1: If a[i] is greater than a[j], exchange the elements of a[i] and a[j] using temp variable else go to Step 3.1.2 STEP 3.1.2: Increment the loop counter j by one. STEP 3.2: Increment the loop counter i by one.

STEP 4: For i=0 and less than n , Repeat the step 4.1 and 4.2 or else if the conditions fails go to Step 5.
STEP 4.1: Printf("%d,",a[i]).
STEP 4.2: Increment the loop counter i by one.
STEP 5: END.

int bin_search(int a[],int n,int length)

STEP 1: START.
STEP 2: Initialize the variables left=0, right length-1, mid.
STEP 3: mid=(left+right)/2.
STEP 4: While left is less than or equal right ,Repeat the step 4.1,4.2,4.3 and 4.4 or else if the condition fails go to step 5.
STEP 4.1: Do mid=(left+right)/2 .
STEP 4.2: If a[mid] is equal to n, Then return mid+1 or else go to Step 4.3.
STEP 4.3: If n greater than a[mid] then do left= mid+1 or else go to step 4.4
STEP 4.4: Do right= mid-1
STEP 5: Return 0.
STEP 6: END.

PROGRAM:

```
#include<stdio.h>
int sel_sort(int a[],int);
int bin_search(int a[],int,int);
void main()
{
   int n;
   printf("Enter the size of the array :");
   scanf("%d",&n);
   int a[n],i;
   printf("Enter the elements of the array seperated by a comma:");
         for(i=0;i<n;i++)
   {
      scanf("%d,",&a[i]);
   }
   printf("\nThe sorted array is :");
   sel_sort(a,n);
   int num;
   printf("\nEnter a number to be searched:");
   scanf("%d",&num);
         if(bin_search(a,num,n)==0)
         {
            printf("Number not found.");
         }
         else
   printf("The number is at %d position", bin_search(a,num,n));
}
int sel_sort(int a[],int n)
{
```

```
    int temp;
    for(int i=0;i<n;i++)
    {
      for(int j=i;j<n;j++)
      {
         if(a[i]>a[j])
         {
            temp=a[j];
            a[j]=a[i];
            a[i]=temp;
         }
      }
    }
    for(int i=0;i<n;i++)
    {
       printf("%d,",a[i]);
    }
}
int bin_search(int a[],int n,int length)
{
    int left,right,mid;
    left=0;
    right=length-1;
    mid=(left+right)/2;
    while(left<=right)
    {
       mid=(left+right)/2;
       if(a[mid]==n)
       {
                              return mid+1;
       }
       else if(n>a[mid])
       {
          left=mid+1;
       }
       else
       right=mid-1;
    }
          return 0;
}
```

RESULT: The array is sorted and number is searched in the array using binary search method.

INPUT:	3. 93,11,41. 39.

OUTPUT:	Enter the size of the array :3 Enter the elements of the array seperated by a comma:93,11,41 The sorted array is :11,41,93, Enter a number to be searched:39 Number not found.