Data Structure and Algorithm
Lab Experiment

Linear and Binary Search & Sorting

Name: Om Ashish Mishra
Reg. No.: 16BCE0789
Slot: G2

Pseudocode:

```
First we declare the header files.
Then we declare the functions linear, binary, sort.
Then we define the main function to take the user's choice about linear or binary operation.
Then we call linear function to get the array and find the element to be searched.
Then we call binary function to get the no of elements, the elements and the element to be searched
Then we call the sort function to sort the array using bubble sort and then we do binary search operation to find the desired element.
If we find the desired element then we print Found and position of the element in both the cases.
Else we print Not Found.
```

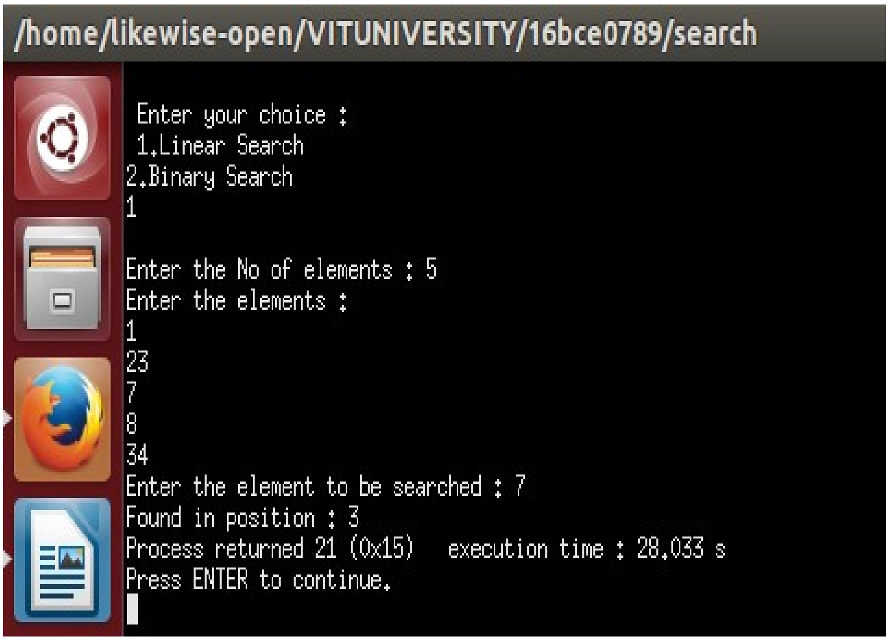
Code:

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

void linear();
void binary();
void sort(int a[50],int n,int d);

void main()
{
   int ch;
   printf("\n Enter your choice : \n 1.Linear Search \n2.Binary Search");
   scanf("%d",&ch);
   switch(ch)
   {
      case 1:
      linear();
      break;

      case 2:
      binary();
      break;

      default:
      printf("Wrong choice");
   }
}
void linear()
{
   int n,a[50],d,i;
   printf("\nEnter the No of elements : ");
   scanf("%d",&n);
   printf("Enter the elements : ");
   for(i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }
   printf("Enter the element to be searched : ");
   scanf("%d",&d);
   int k=0,j=0;
   for(i=0;i<n;i++)
   {
      if(d==a[i])
      {
         k=1;
         j=(i+1);
         break;
      }
   }
   if(k==1)
   {
      printf("Found in position : %d",j);
   }
   else
   {
      printf("Not found");
   }
}

void binary()
{
   int n,a[50],d,i;
   printf("\nEnter the No of elements : ");
   scanf("%d",&n);
   printf("Enter the elements : ");
   for(i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }
   printf("Enter the element to be searched : ");
   scanf("%d",&d);
   sort(a,n,d);
}

void sort(int a[50],int n, int d)
{
   int i,j,temp;
   for(i=0;i<n-1;i++)
   {
      for(j=0;j<n-i-1;j++)
      {
         if(a[j]>a[j+1])
         {
            temp=a[j];
            a[j]=a[j+1];
            a[j+1]=temp;
         }
      }
   }
   printf("Sorted array : \n");
   for(i=0;i<n;i++)
   {
```

```
            printf("%d\t",a[i]);
        }
        printf("\n");
        int l=0,u=(n-1),k=0;
        int mid=(l+u)/2;
        while(l<=u)
        {
            if(d==a[mid])
            {
                k=1;
                j=(mid+1);
                break;
            }
            if(a[mid]<d)
            {
                u=mid-1;
            }
            if(a[mid]>d)
            {
                l=mid+1;
            }
        }
        if(k==1)
        {
            printf("Element is found at : %d",j);
        }
        else
        {
            printf("Not found");
        }
    }
```

Output:

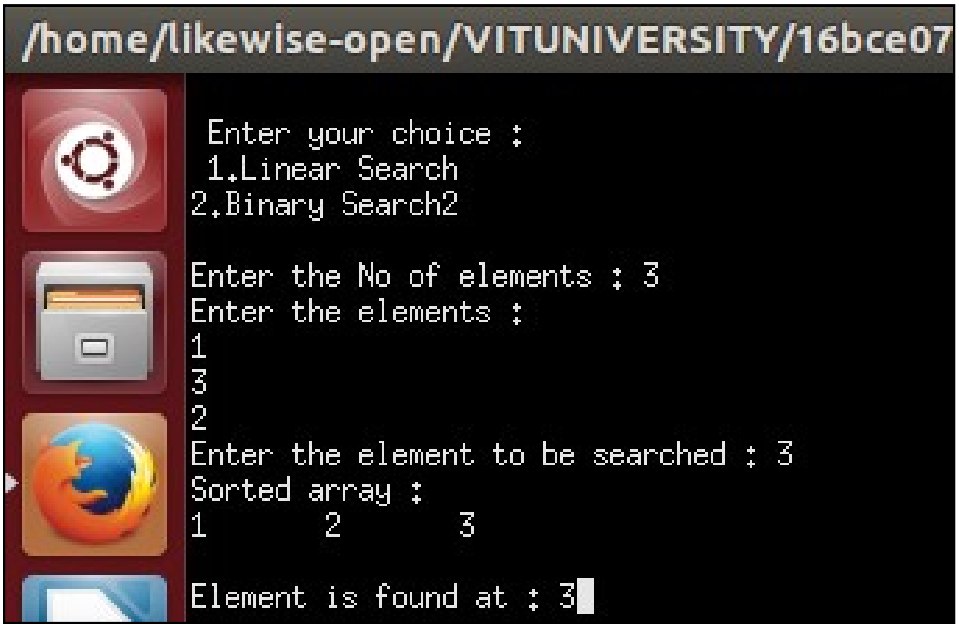
Linear Sort:



/home/likewise-open/VITUNIVERSITY/16bce0789/search

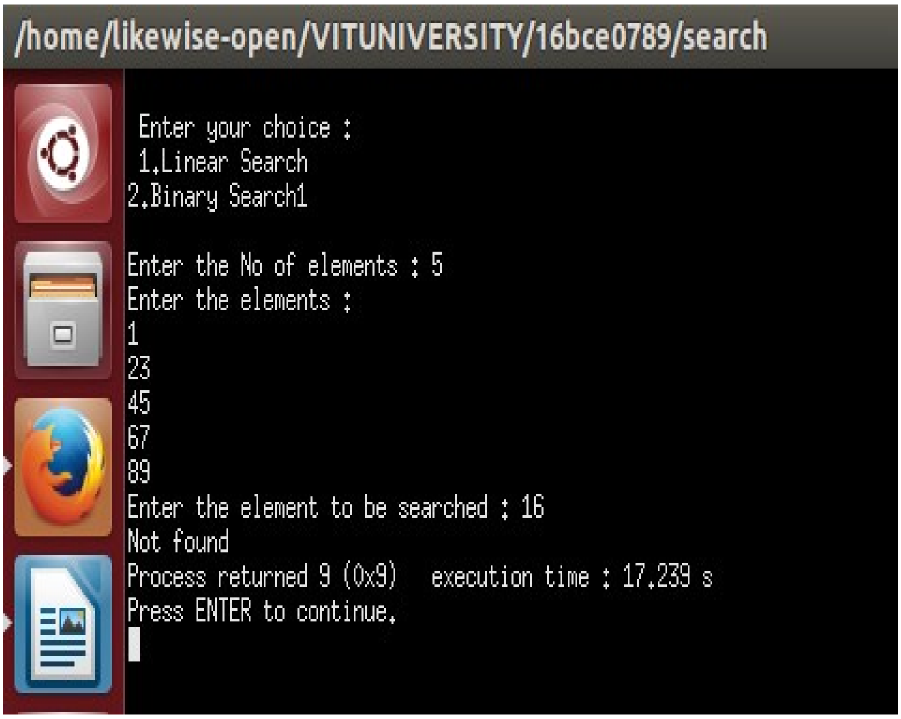
```
 Enter your choice :
 1.Linear Search
2.Binary Search
1

Enter the No of elements : 5
Enter the elements :
1
23
7
8
34
Enter the element to be searched : 7
Found in position : 3
Process returned 21 (0x15)   execution time : 28.033 s
Press ENTER to continue.
```

Binary Sort:



```
 Enter your choice :
 1.Linear Search
2.Binary Search2

Enter the No of elements : 3
Enter the elements :
1
3
2
Enter the element to be searched : 3
Sorted array :
1       2       3

Element is found at : 3
```

Not Found:



```
 Enter your choice :
 1.Linear Search
2.Binary Search1

Enter the No of elements : 5
Enter the elements :
1
23
45
67
89
Enter the element to be searched : 16
Not found
Process returned 9 (0x9)   execution time : 17.239 s
Press ENTER to continue.
```

Sorting of the details of a student according to the Registration Number of the student:

Pseudo code**:**

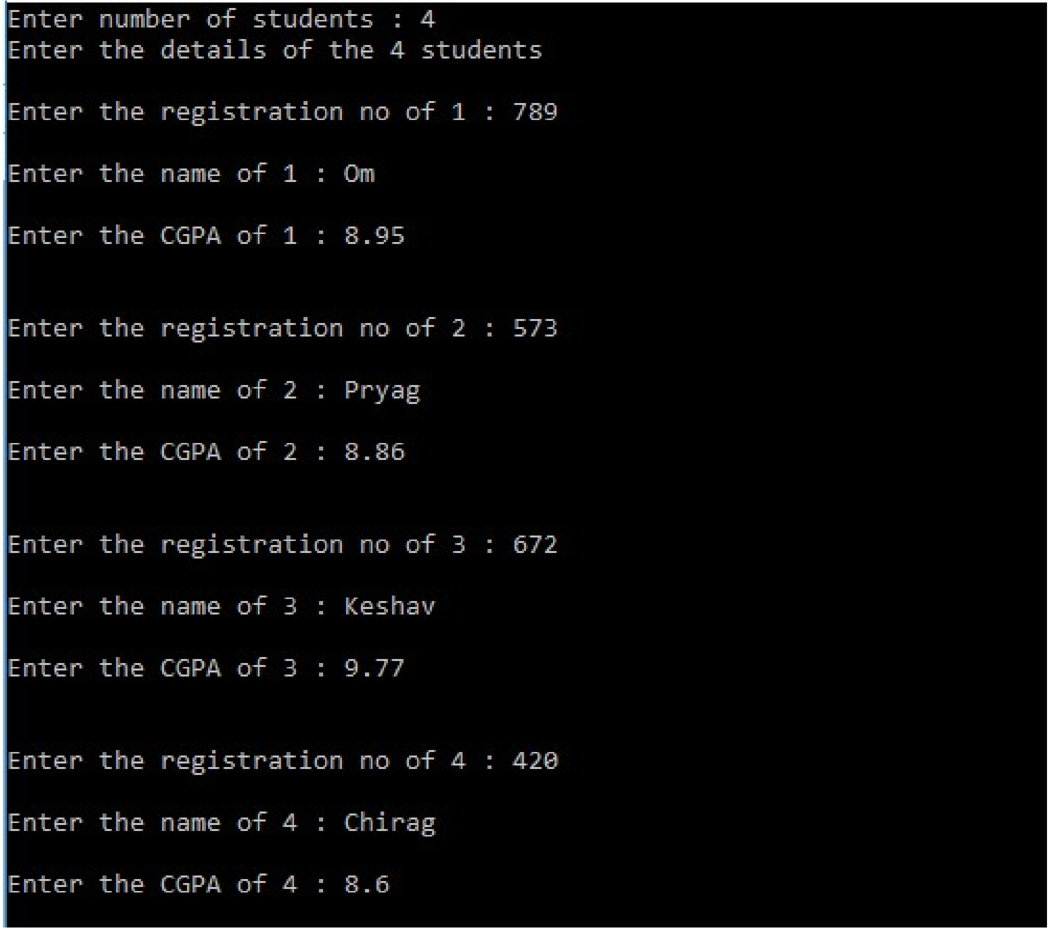
First we declare the header files.
Then we form the structure that contains the details of the student (Registration Number, Name, CGPA).
Then we declare the prototype of the function binary_search.
Then we call the main function and take the details f the student like no of students their Registration Number, Name and CGPA.
Then we call the binary_search function to sort the elements of the structure according to the Registration Number.
Then we print the output of the program in an ascending manner.

Code:

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

struct student
{
   int Reg_No;
   char Name[50];
   float cgpa;
}s[100];

void bubble_sort(int);

int main()
{
  int n,i;
  printf("Enter number of students : ");
  scanf("%d", &n);

  printf("Enter the details of the %d students \n",n);

  for (i = 0; i < n; i++)
  {
    printf("\nEnter the registration no of %d : ",(i+1));
    scanf("%d", &s[i].Reg_No);
    printf("\nEnter the name of %d : ",(i+1));
    scanf("%s",&s[i].Name);
    printf("\nEnter the CGPA of %d : ",(i+1));
    scanf("%f",&s[i].cgpa);
    printf("\n");
  }
  bubble_sort(n);

  printf("Sorted list in ascending order according to registration Number : \n");

  for ( i = 0 ; i < n ; i++ )
  {
    printf("\nThe registration no of %d : ",(i+1));
    printf("%d\n", s[i].Reg_No);
    printf("\nThe name of %d : ",(i+1));
    printf("%s\n",s[i].Name);
    printf("\nThe CGPA of %d : ",(i+1));
    printf("%f\n",s[i].cgpa);
    printf("\n");
  }


  return 0;
}

void bubble_sort(int n)
{
  int i, j, t=0;
  char na[50]="";
  float cp=0;

  for (i = 0 ; i < ( n - 1 ); i++)
  {
    for (j = 0 ; j < n - i - 1; j++)
    {
      if (s[j].Reg_No > s[j+1].Reg_No)
      {
        t = s[j].Reg_No;
        s[j].Reg_No = s[j+1].Reg_No;
        s[j+1].Reg_No = t;

        strcpy(na,s[j].Name);
        strcpy(s[j].Name,s[j+1].Name);
        strcpy(s[j+1].Name,na);

        cp=s[j].cgpa;
        s[j].cgpa=s[j+1].cgpa;
        s[j+1].cgpa=cp;
      }
    }
  }
}
```

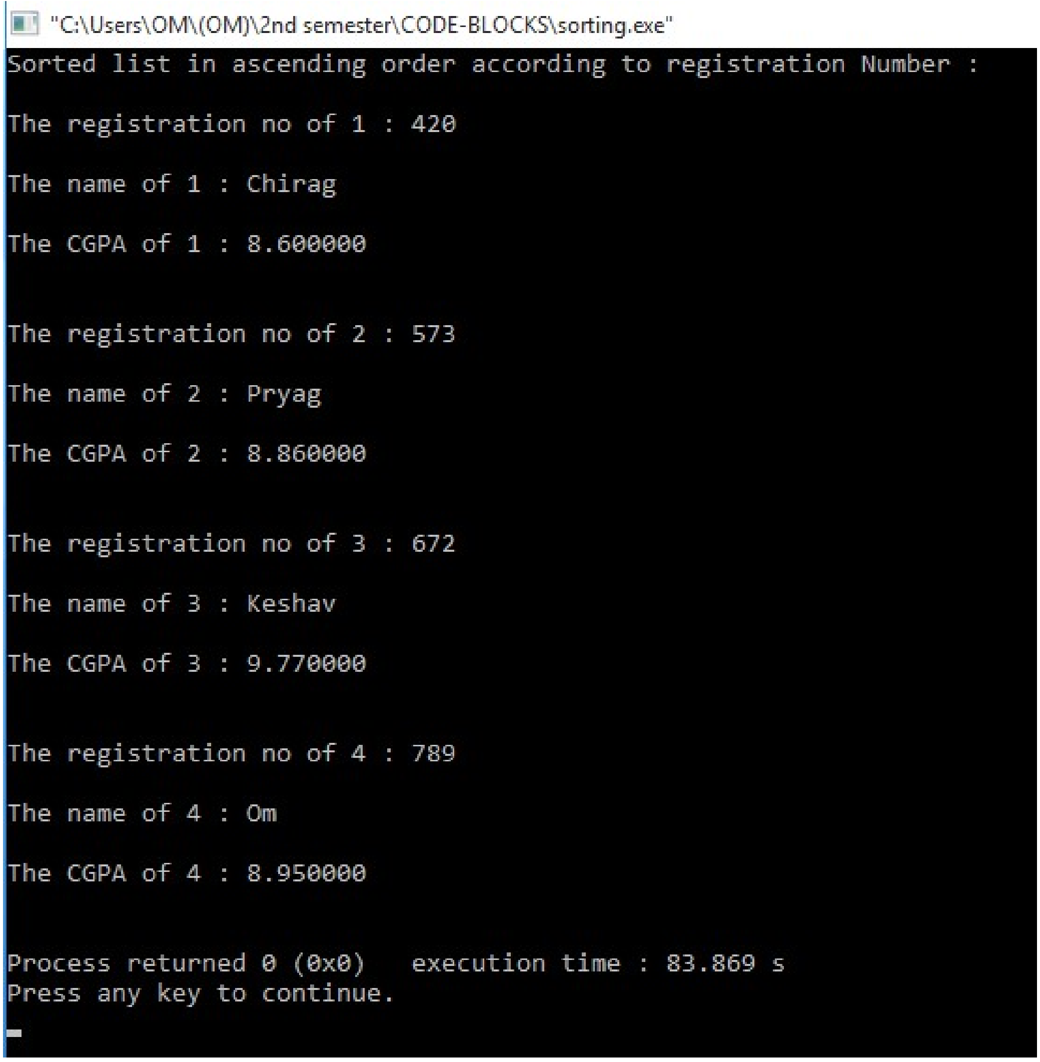
The Output:

The input:



```
Enter number of students : 4
Enter the details of the 4 students

Enter the registration no of 1 : 789

Enter the name of 1 : Om

Enter the CGPA of 1 : 8.95


Enter the registration no of 2 : 573

Enter the name of 2 : Pryag

Enter the CGPA of 2 : 8.86


Enter the registration no of 3 : 672

Enter the name of 3 : Keshav

Enter the CGPA of 3 : 9.77


Enter the registration no of 4 : 420

Enter the name of 4 : Chirag

Enter the CGPA of 4 : 8.6
```

The Output:



"C:\Users\OM\(OM)\2nd semester\CODE-BLOCKS\sorting.exe"

```
Sorted list in ascending order according to registration Number :

The registration no of 1 : 420

The name of 1 : Chirag

The CGPA of 1 : 8.600000


The registration no of 2 : 573

The name of 2 : Pryag

The CGPA of 2 : 8.860000


The registration no of 3 : 672

The name of 3 : Keshav

The CGPA of 3 : 9.770000


The registration no of 4 : 789

The name of 4 : Om

The CGPA of 4 : 8.950000


Process returned 0 (0x0)   execution time : 83.869 s
Press any key to continue.
```

Thank You