```
/*
                                    ASSIGNMENT NO. 1
      NAME- ABRAR SHAIKH                                        ROLL NO. -
23570
      TOPIC- BUBBLE SORT, INSERTION SORT, QUICK SORT, BINARY SEARCH,
LINEAR SEARCH
*/


#include <iostream>
#include <string>


using namespace std;


//structure for storing data
struct Student
{
      int rollno;
      string name;
      float sgpa;
};


//function for displaying data
void display(Student *s[],int c)
{
      cout<<"*************Displaying student
details*****************"<<endl;
      for(int i=0; i<c; i++)
      {
            cout<<"Roll No.:"<<s[i]->rollno<<"\n"<<"Name:"<<s[i]-
>name<<"\n"<<"SGPA:"<<s[i]->sgpa<<endl<<endl;
      }
}
```

```
//function for bubble sort... arranging elements in ascending order of
roll no.
void bSort(Student *s[],int n)
{
	int flag;

	//creating temporary pointer for swapping
	Student *temp;

	//outer loop for number of passes
	for(int i=0; i<n-1; i++)
	{
		flag=0;

		//inner loop for swapping elements
		for(int j=0; j<n-i-1; j++)
		{
			if(s[j]->rollno>s[j+1]->rollno)
			{
				temp=s[j];
				s[j]=s[j+1];
				s[j+1]=temp;
			}
		}
	}

	cout<<endl<<"Ascending order of roll numbers"<<endl;
	display(s,n);
}

//function for insertion sort... arranging elements in alphabetical order
void insertionSort(Student *s[], int n)
```

```
{
	//temporary pointer
	Student *temp;
	string k;
	int j;

	//outer loop for number of passes
	for(int i=1; i<n; i++)
	{
		//storing value at i in temporary pointer
		temp=s[i];
		//copying name at ith position in k
		k=s[i]->name;

		//inner loop for swapping
		for(j=i-1; (j>-1)&&(k<s[j]->name); j--)
		{
			s[j+1]=s[j];
		}
		//placing value at sorted position
		s[j+1]=temp;
	}

	cout<<"List of students arranged alphabetically"<<endl;
	display(s,n);
}

//function for linear search... printing student info based on sgpa
void linearSearch(Student *s[], int n, float key)
{
	cout<<"SR NO."<<"\t"<<"Roll No."<<"\t"<<"Name"<<"\t"<<"SGPA"<<endl;
	for(int i=0; i<n; i++)
```

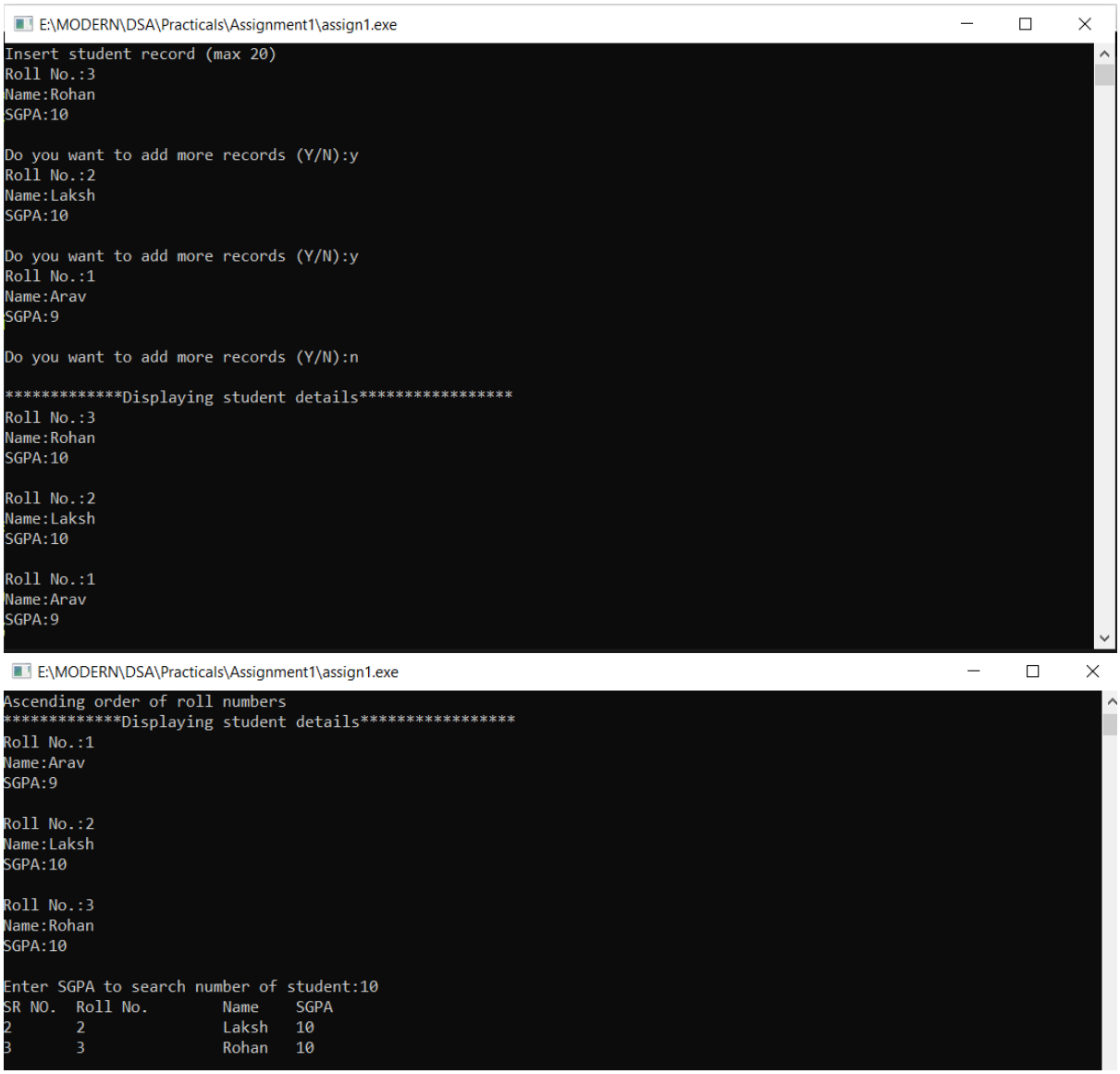
```
      {
            if(key==s[i]->sgpa)
            {
                  cout<<i+1<<"\t"<<s[i]->rollno<<"\t\t"<<s[i]-
>name<<"\t"<<s[i]->sgpa<<endl;
            }
      }
}

// Function for partition of array into subarray
int partition(Student *a[], int l, int u)
{
    Student *pivot = a[l];  // Choosing the first element as the pivot
    float pivotValue = pivot->sgpa;
    int start = l;
    int end = u;

    while (start < end)
    {
        // Move start index to the right, while elements are greater than
or equal to pivot
        while (start < u && a[start]->sgpa >= pivotValue)
        {
            start++;
        }
        // Move end index to the left, while elements are less than pivot
        while (a[end]->sgpa < pivotValue)
        {
            end--;
        }
        // Swap elements if necessary
        if (start < end)
        {
```

```
            swap(a[start], a[end]);
        }
    }
    // Swap the pivot element with the element at end index
    swap(a[l], a[end]);
    return end;  // Return the position of the pivot
}

//function for quick sort
void quickSort(Student *s[], int l, int u)
{
 if (l< u)
 {
   int loc = partition(s, l, u);
   quickSort(s, l, loc - 1);
   quickSort(s, loc + 1, u);
 }
}

// Function for binary search by name (non-recursive)
void binarySearchByName(Student *s[], int n, const string &key)
{
    insertionSort(s,n);
    int low = 0;
    int high = n - 1;
    bool found = false;

    while (low <= high)
    {
        int mid = low + (high - low) / 2;
        // Compare the middle student's name with the search key
        if (s[mid]->name < key)
```

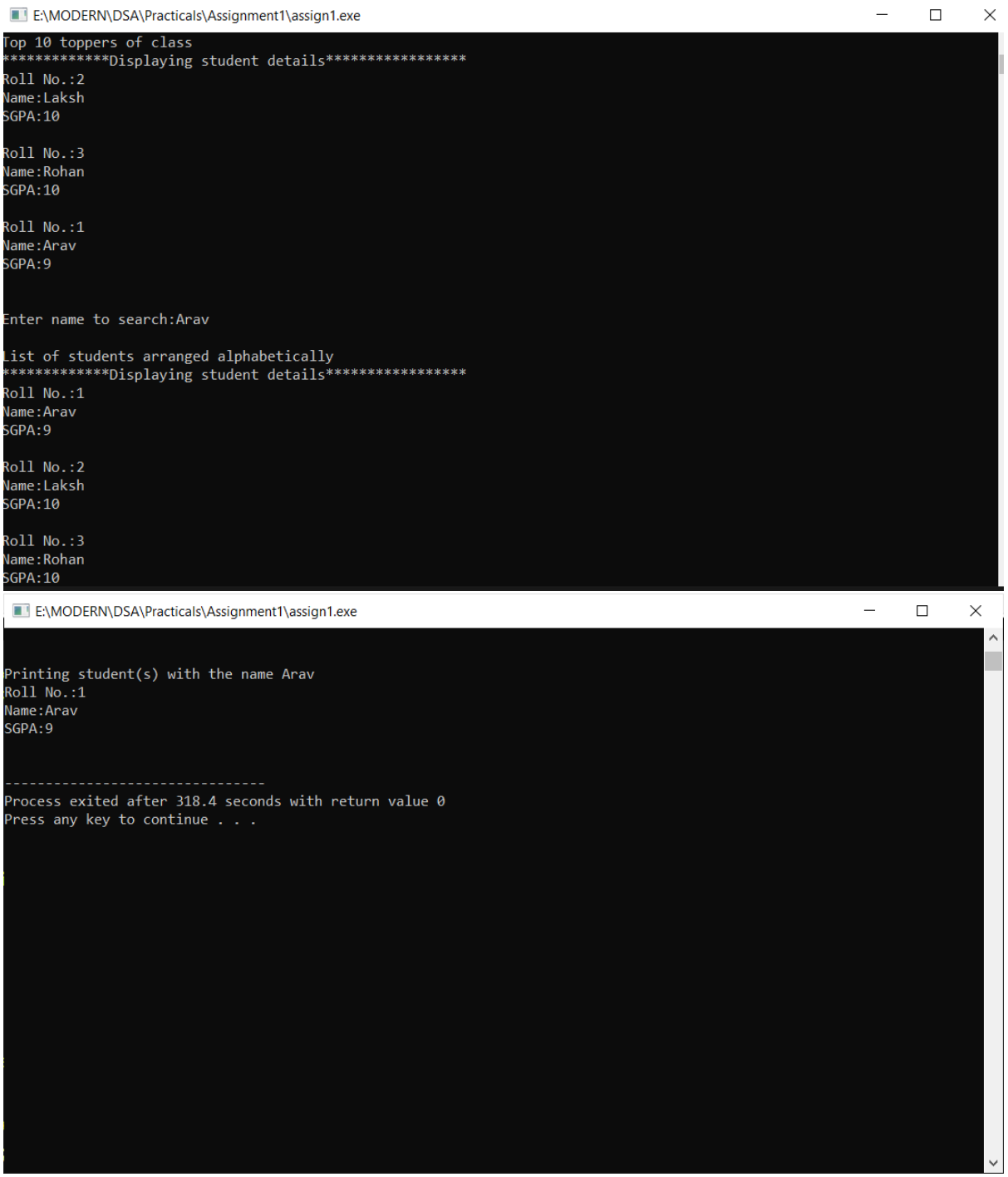
```
        {
            low = mid + 1;
        }
        else if (s[mid]->name > key)
        {
            high = mid - 1;
        }
        else
        {
            // Found the first occurrence, print all occurrences
            found = true;
            // Print all students with the same name, including duplicates
            cout<<endl<<"Printing student(s) with the name "<<key<<endl;
            int i = mid;
            while (i >= 0 && s[i]->name == key)
            {
                cout << "Roll No.:" << s[i]->rollno << "\n" << "Name:" <<
s[i]->name << "\n" << "SGPA:" << s[i]->sgpa << endl << endl;
                i--;
            }
            i = mid + 1;
            while (i < n && s[i]->name == key)
            {
                cout << "Roll No.:" << s[i]->rollno << "\n" << "Name:" <<
s[i]->name << "\n" << "SGPA:" << s[i]->sgpa << endl << endl;
                i++;
            }
            break;
        }
    }
    if (!found)
    {
        cout << "No student found with the name " << key << endl;
```

```
    }
}

int main()
{
      int c=0;
      Student *s[20];

      //loop for inserting data into structure
      cout<<"Insert student record (max 20)"<<endl;
      for(int i=0; i<20; i++)
      {
            s[i]=new Student;

            cout<<"Roll No.:";
            cin>>s[i]->rollno;
            cout<<"Name:";
            cin>>s[i]->name;
            cout<<"SGPA:";
            cin>>s[i]->sgpa;

            //for counting number of entries
            c++;

            char ch;
            cout<<endl<<"Do you want to add more records (Y/N):";
            cin>>ch;

            if(ch=='N'||ch=='n')
                  break;
      }
```

```
        cout<<endl;

        //function for displaying data in the structure
        display(s,c);

        //function for bubble sort
        bSort(s,c);

        float key;
        cout<<"Enter SGPA to search number of student:";
        cin>>key;
        //function for sgpa
        linearSearch(s,c,key);

        // Call quickSort to sort the array
        cout<<endl<<"Top 10 toppers of class"<<endl;
        quickSort(s, 0, c-1);
        display(s,c);

        string n;
        cout<<endl<<"Enter name to search:";
        cin>>n;
        cout<<endl;

        //call binarysearch to serach by name
        binarySearchByName(s, c, n);

        return 0;
}
```


E:\MODERN\DSA\Practicals\Assignment1\assign1.exe
Insert student record (max 20)
Roll No.:3
Name:Rohan
SGPA:10
Do you want to add more records (Y/N):y
Roll No.:2
Name:Laksh
SGPA:10
Do you want to add more records (Y/N):y
Roll No.:1
Name:Arav
SGPA:9
Do you want to add more records (Y/N):n
**********Displaying student details**********
Roll No.:3
Name:Rohan
SGPA:10
Roll No.:2
Name:Laksh
SGPA:10
Roll No.:1
Name:Arav
SGPA:9

E:\MODERN\DSA\Practicals\Assignment1\assign1.exe
Ascending order of roll numbers
**********Displaying student details**********
Roll No.:1
Name:Arav
SGPA:9
Roll No.:2
Name:Laksh
SGPA:10
Roll No.:3
Name:Rohan
SGPA:10
Enter SGPA to search number of student:10
SR NO. Roll No. Name SGPA
2 2 Laksh 10
3 3 Rohan 10


E:\MODERN\DSA\Practicals\Assignment1\assign1.exe
Top 10 toppers of class
**********Displaying student details**********
Roll No.:2
Name:Laksh
SGPA:10
Roll No.:3
Name:Rohan
SGPA:10
Roll No.:1
Name:Arav
SGPA:9
Enter name to search:Arav
List of students arranged alphabetically
**********Displaying student details**********
Roll No.:1
Name:Arav
SGPA:9
Roll No.:2
Name:Laksh
SGPA:10
Roll No.:3
Name:Rohan
SGPA:10
E:\MODERN\DSA\Practicals\Assignment1\assign1.exe
Printing student(s) with the name Arav
Roll No.:1
Name:Arav
SGPA:9

Process exited after 318.4 seconds with return value 0
Press any key to continue . . .