```
#include<iostream>
#include<string.h>
using namespace std;
//class declaration
class Student_class
{
private:
     //data members
     struct student
     {
          int roll_no;
          string name;
          int credit[5];
          int grade[5];
          int sgpa;
     }s[20];
public:
     //method declaration
     void input(int);
     bool name_validation(string);//name validation
     bool rollNo_validation(int);//roll number validation
     void display(int);//display method
     void sort_rollNo(int);//bubble sort
     void sort_aplhabetically(int);//insertion sort
     void sort_sgpa(int,int);//quick sort
     void display_topper(int);//to display toppers
     void search_SGPA(int);//linear search
     void search_name(int);//binary search
```

```
};
//definition of input method
void Student_class::input(int n)
{
     for(int i=0;i<n;i++)
     {
          bool valid;
          cout<<endl<<"-----Student"<<i+1<<"----"<<endl;
          do
          {
               cout<<"Roll_no.:";
               cin>>s[i].roll_no;
               valid=rollNo_validation(i); //roll number validation
          }while(!valid);
          cin.ignore();
          do
          {
               cout<<"Name:\n";
               getline(cin,s[i].name);
               valid=name_validation(s[i].name); //name validation
          }while(!valid);
          cout<<"Enter marks of 5 subjects:"<<endl;
          double sum_of_product=0;//temporary variable to calculate SGPA
          int total_credit=0;
          for(int j=0;j<5;j++)
          {
               cout<<"**Subject "<<j+1<<":**"<<endl;
               do
               {
```

```
                cout<<"Credit:";
                cin>>s[i].credit[j];
                if(s[i].credit[j]>5||s[i].credit[j]<1)
                    cout<<"Credit should be in range of 1 to
5"<<endl;//validating credit
            }while(s[i].credit[j]>5||s[i].credit[j]<1);

            total_credit+=s[i].credit[j];
            do
            {
                cout<<"Grade:"<<endl;
                cin>>s[i].grade[j];
                if(s[i].grade[j]>10||s[i].grade[j]<1)
                    cout<<"Grade should be in range 1 to
10"<<endl;//validating grade
            }while(s[i].grade[j]>10||s[i].grade[j]<1);

            sum_of_product+=(s[i].credit[j]*s[i].grade[j]);
        }
        s[i].sgpa=sum_of_product/total_credit; //calculating SGPA

    }
}
//definition of method to validate roll number
bool Student_class::rollNo_validation(int i)
{
    if(s[i].roll_no<0)
    {
        cout<<"Roll number should not be negative"<<endl;
```

```
            return false; //if it matches return false
        }
        for(int j=i-1;j>=0;j--)
        {
            if(s[i].roll_no==s[j].roll_no)//comparing roll number with all
previous roll numbers
            {
                cout<<"Roll number should be unique"<<endl;
                return false; //if it matches return false
            }
        }
        return true;
}
//definition of method to validate name
bool Student_class::name_validation(string name)
{
        int i=0,count=0;
        while(name[i]!='\0')
        {
            if(isspace(name[i])) //validation for full name
                count++;
            i++;
        }
        if(count==2)
        {
            i=0;
            while(name[i]!='\0')
            {
```

```
            if(isalpha(name[i])||isspace(name[i])) //validation for correct
name
                i++;
            else
                break;
        }
        if(name[i]=='\0')
            return true;
    }
    cout<<"Enter valid name"<<endl;
    return false;
}
//definition of display
void Student_class::display(int i)
{

    cout<<i+1<<"\t"<<s[i].roll_no<<"\t"<<s[i].name<<"\t"<<s[i].sgpa<<endl;
}
//definition of sort according to roll number
void Student_class::sort_rollNo(int n) //bubble sort
{
    bool swapped;
    student temp;
        for(int i=0;i<n;i++)
        {
            swapped=false;
            for(int j=0;j<n-i-1;j++)
            {
                if(s[j].roll_no>s[j+1].roll_no)
```

```
                    {
                         temp=s[j];
                         s[j]=s[j+1];
                         s[j+1]=temp;
                         swapped=true;
                    }
               }
               cout<<"Pass "<<i+1<<":"<<endl; //display pass wise output
               for(int k=0;k<n;k++)
                    display(k);
               if(!swapped)
                    break;
          }
          cout<<"List sorted successfully"<<endl;
}
//definition of sort alphabetically
void Student_class::sort_aplhabetically(int n) //insertion sort
 {
     int i,j;
     student temp1;
     for(i=1;i<n;i++)
     {
          temp1=s[i];
          j=i-1;
          while(j>=0&&s[j].name>temp1.name){
               s[j+1]=s[j];
               j--;
          }
          s[j+1]=temp1;
```

```
          //pass wise display output
          cout<<"Pass "<<i<<":"<<endl;
          for(int k=0;k<n;k++)
               display(k);
          }
 }
//definition of search using SGPA
void Student_class::search_SGPA(int n) //linear search
{
     double key;
     bool found=false;
     cout<<"Enter SGPA to be search:";
     cin>>key;
     for(int i=0;i<n;i++)
     {
          if(s[i].sgpa==key)
          {
               display(i);
               found=true;
          }

     }
     if(!found)
          cout<<"No student with SGPA "<<key<<" found"<<endl;
}
//definition of search according to name
void Student_class::search_name(int n) //binary search
{
     string key;
```

```
    cin.ignore();
    cout<<"Enter search key:";
    getline(cin,key);
    int low=0,high=n-1,mid;
    bool found=false;
    while(low<=high)
    {
        mid=low+(high-low)/2;
        int x=s[mid].name.compare(key);
        if(x==0)
        {
            found=true;
            display(mid);
            break;
        }
        else if(x>0)
            high=mid-1;
        else
            low=mid+1;
    }
    if(!found)
        cout<<"Student with name '"<<key<<"' not found"<<endl;


}
//definition of sort with SGPA method
void Student_class::sort_sgpa(int left,int right)
{
    static int pass=0; //to display pass wise output
```

```
static int n=right+1;
//quick sort
if(left>=right)
     return;
int i=left;
int j=right+1;
student pivot=s[left];
while(1)
{
     do{
          i++;
     }while(s[i].sgpa<pivot.sgpa);

     do{
          j--;
     }while(s[j].sgpa>pivot.sgpa);
     if(i>=j)
          break;
     else{
          student temp=s[i];
          s[i]=s[j];
          s[j]=temp;
     }
}
s[left]=s[j];
s[j]=pivot;
//pass wise output
cout<<"Pass "<<pass++<<":"<<endl;
for(int k=0;k<n;k++)
```

```
        display(k);
    sort_sgpa(left,j-1); //recursion
    sort_sgpa(j+1,right);

}

//definition of function to display topper
void Student_class::display_topper(int n)
{
    int top_num;
    sort_sgpa(0,n-1); //sort according to SGPA
    cout<<"Enter number of toppers to be display:";
    cin>>top_num; //total number of toppers to be displayed
    if(top_num>n)
        cout<<"only "<<n<<" student's records are  available"<<endl;
    else
        for(int i=n-1;i>=n-top_num;i--) //quick sort will sort in ascending
order so toppers should be display according to descending order
            display(i);
}
//driver function
int main()
{
    Student_class obj;
    int n,choice;
    do
    {
        cout<<"Enter valid number of students:";
        cin>>n; //Enter total number of students
```

```
    }while(n<=0||n>20);
    obj.input(n);
    do
    {
        //display menus to user
        cout<<"-------------------------------------------------------------
"<<endl;
        cout<<"1:Display\n2:Sort list according to roll number\n3:Sort list
alphabetically\n4:Sort with SGPA\n5:Search student with SGPA\n6:Search
student according to name\n7:Display toppers\n8:Exit"<<endl;
        cout<<"Enter choice:";
        cin>>choice;//enter choice of user
        cout<<"-------------------------------------------------------------
"<<endl;
        switch(choice)
        {
        case 1: //to display all records
            cout<<"SrNo\tRoll No\t\tName\tSGPA"<<endl;
            for(int i=0;i<n;i++)
                obj.display(i);
            break;
        case 2: //sort student's record according to roll number
            obj.sort_rollNo(n);
            break;
        case 3:
            obj.sort_aplhabetically(n); //sort according to name
            cout<<"Sorted Successfully"<<endl;
            break;
        case 4:
```

```
            obj.sort_sgpa(0,n-1); //sort according to SGPA
            cout<<"Sorted Successfully"<<endl;
            break;
        case 5:
            obj.search_SGPA(n); //search a student with SGPA
            break;
        case 6://search a Student with name (binary search)
            obj.sort_aplhabetically(n);
            obj.search_name(n);
            break;
        case 7://to display list of toppers
            obj.display_topper(n);
            break;
        case 8:
            cout<<"Thank You"<<endl;
            break;
        default:
            cout<<"Enter valid choice"<<endl;
        }
    }while(choice!=8);
    return 0;
}
```

Output:

```
Enter valid number of students:-5
Enter valid number of students:0
Enter valid number of students:22
Enter valid number of students:5

-----Student1----
Roll_no.:101
Name:
abcd abcd abcd
Enter marks of 5 subjects:
**Subject 1:**
Credit:5
Grade:
10
**Subject 2:**
Credit:5
Grade:
10
**Subject 3:**
Credit:5
Grade:
10
**Subject 4:**
Credit:6
Credit should be in range of 1 to 5
Credit:4
Grade:
```

```
9
**Subject 5:**
Credit:4
Grade:
19
Grade should be in range 1 to 10
Grade:
9

-----Student2----
Roll_no.:103
Name:
pqrs pqrs pqrs
Enter marks of 5 subjects:
**Subject 1:**
Credit:3
Grade:
7
**Subject 2:**
Credit:3
Grade:
7
**Subject 3:**
Credit:2
Grade:
7
**Subject 4:**
Credit:3
Grade:
```

```
8
**Subject 5:**
Credit:3
Grade:
7

-----Student3----
Roll_no.:101
Roll number should be unique
Roll_no.:104
Name:
wxyz wz34
Enter valid name
Name:
wxyz wxyz
Enter valid name
Name:
wxyz wxyz wxyz
Enter marks of 5 subjects:
**Subject 1:**
Credit:4
Grade:
8
**Subject 2:**
Credit:4
Grade:
8
**Subject 3:**
Credit:4
```

```
Grade:
8
**Subject 4:**
Credit:4
Grade:
8
**Subject 5:**
Credit:4
Grade:
8

-----Student4----
Roll_no.:110
Name:
efgh efgh efgh
Enter marks of 5 subjects:
**Subject 1:**
Credit:2
Grade:
7
**Subject 2:**
Credit:2
Grade:
6
**Subject 3:**
Credit:1
Grade:
6
**Subject 4:**
```

```
Credit:3
Grade:
7
**Subject 5:**
Credit:2
Grade:
7

-----Student5----
Roll_no.:105
Name:
hijk hijk hijk
Enter marks of 5 subjects:
**Subject 1:**
Credit:5
Grade:
10
**Subject 2:**
Credit:5
Grade:
10
**Subject 3:**
Credit:5
Grade:
10
**Subject 4:**
Credit:5
Grade:
10
```

```
**Subject 5:**
Credit:5
Grade:
10
-------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
7:Display toppers
8:Exit
Enter choice:1
-------------------------------------------------------------
SrNo Roll No      Name SGPA
1    101 abcd abcd abcd   9
2    103 pqrs pqrs pqrs   7
3    104 wxyz wxyz wxyz   8
4    110 efgh efgh efgh   6
5    105 hijk hijk hijk   10
-------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
7:Display toppers
```

```
8:Exit
Enter choice:2
------------------------------------------------------------
Pass 1:
1    101 abcd abcd abcd   9
2    103 pqrs pqrs pqrs   7
3    104 wxyz wxyz wxyz   8
4    105 hijk hijk hijk   10
5    110 efgh efgh efgh   6
Pass 2:
1    101 abcd abcd abcd   9
2    103 pqrs pqrs pqrs   7
3    104 wxyz wxyz wxyz   8
4    105 hijk hijk hijk   10
5    110 efgh efgh efgh   6
List sorted successfully
------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
7:Display toppers
8:Exit
Enter choice:3
------------------------------------------------------------
Pass 1:
1    101 abcd abcd abcd   9
```

```
2    103 pqrs pqrs pqrs   7
3    104 wxyz wxyz wxyz   8
4    105 hijk hijk hijk   10
5    110 efgh efgh efgh   6
Pass 2:
1    101 abcd abcd abcd   9
2    103 pqrs pqrs pqrs   7
3    104 wxyz wxyz wxyz   8
4    105 hijk hijk hijk   10
5    110 efgh efgh efgh   6
Pass 3:
1    101 abcd abcd abcd   9
2    105 hijk hijk hijk   10
3    103 pqrs pqrs pqrs   7
4    104 wxyz wxyz wxyz   8
5    110 efgh efgh efgh   6
Pass 4:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    105 hijk hijk hijk   10
4    103 pqrs pqrs pqrs   7
5    104 wxyz wxyz wxyz   8
Sorted Successfully
---------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
```

```
6:Search student according to name
7:Display toppers
8:Exit
Enter choice:3
------------------------------------------------------------
Pass 1:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    105 hijk hijk hijk   10
4    103 pqrs pqrs pqrs   7
5    104 wxyz wxyz wxyz   8
Pass 2:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    105 hijk hijk hijk   10
4    103 pqrs pqrs pqrs   7
5    104 wxyz wxyz wxyz   8
Pass 3:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    105 hijk hijk hijk   10
4    103 pqrs pqrs pqrs   7
5    104 wxyz wxyz wxyz   8
Pass 4:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    105 hijk hijk hijk   10
4    103 pqrs pqrs pqrs   7
5    104 wxyz wxyz wxyz   8
```

```
Sorted Successfully
------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
7:Display toppers
8:Exit
Enter choice:4
------------------------------------------------------------
Pass 0:
1    103 pqrs pqrs pqrs   7
2    110 efgh efgh efgh   6
3    104 wxyz wxyz wxyz   8
4    101 abcd abcd abcd   9
5    105 hijk hijk hijk   10
Pass 1:
1    110 efgh efgh efgh   6
2    103 pqrs pqrs pqrs   7
3    104 wxyz wxyz wxyz   8
4    101 abcd abcd abcd   9
5    105 hijk hijk hijk   10
Sorted Successfully
------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
```

```
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
7:Display toppers
8:Exit
Enter choice:5
------------------------------------------------------------
Enter SGPA to be search:8
3   104  wxyz wxyz wxyz   8
------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
7:Display toppers
8:Exit
Enter choice:5
------------------------------------------------------------
Enter SGPA to be search:1
No student with SGPA 1 found
------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
```

```
7:Display toppers
8:Exit
Enter choice:6
-------------------------------------------------------------
Pass 1:
1    110 efgh efgh efgh   6
2    103 pqrs pqrs pqrs   7
3    104 wxyz wxyz wxyz   8
4    101 abcd abcd abcd   9
5    105 hijk hijk hijk   10
Pass 2:
1    110 efgh efgh efgh   6
2    103 pqrs pqrs pqrs   7
3    104 wxyz wxyz wxyz   8
4    101 abcd abcd abcd   9
5    105 hijk hijk hijk   10
Pass 3:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    103 pqrs pqrs pqrs   7
4    104 wxyz wxyz wxyz   8
5    105 hijk hijk hijk   10
Pass 4:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    105 hijk hijk hijk   10
4    103 pqrs pqrs pqrs   7
5    104 wxyz wxyz wxyz   8
Enter search key:pqrs pqrs pqrs
```

```
4    103 pqrs pqrs pqrs   7
------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
7:Display toppers
8:Exit
Enter choice:6
------------------------------------------------------------
Pass 1:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    105 hijk hijk hijk   10
4    103 pqrs pqrs pqrs   7
5    104 wxyz wxyz wxyz   8
Pass 2:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    105 hijk hijk hijk   10
4    103 pqrs pqrs pqrs   7
5    104 wxyz wxyz wxyz   8
Pass 3:
1    101 abcd abcd abcd   9
2    110 efgh efgh efgh   6
3    105 hijk hijk hijk   10
4    103 pqrs pqrs pqrs   7
```

```
5   104 wxyz wxyz wxyz   8
Pass 4:
1   101 abcd abcd abcd   9
2   110 efgh efgh efgh   6
3   105 hijk hijk hijk   10
4   103 pqrs pqrs pqrs   7
5   104 wxyz wxyz wxyz   8
Enter search key:acde cded cdes
Student with name 'acde cded cdes' not found
------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
7:Display toppers
8:Exit
Enter choice:7
------------------------------------------------------------
Pass 2:
1   103 pqrs pqrs pqrs   7
2   110 efgh efgh efgh   6
3   104 wxyz wxyz wxyz   8
4   101 abcd abcd abcd   9
5   105 hijk hijk hijk   10
Pass 3:
1   110 efgh efgh efgh   6
2   103 pqrs pqrs pqrs   7
```

```
3    104  wxyz wxyz wxyz   8
4    101  abcd abcd abcd   9
5    105  hijk hijk hijk   10
Enter number of toppers to be display:3
5    105  hijk hijk hijk   10
4    101  abcd abcd abcd   9
3    104  wxyz wxyz wxyz   8
------------------------------------------------------------
1:Display
2:Sort list according to roll number
3:Sort list alphabetically
4:Sort with SGPA
5:Search student with SGPA
6:Search student according to name
7:Display toppers
8:Exit
Enter choice:8
------------------------------------------------------------
Thank You
```