Problem Statement : Write C/C++ program to store marks scored for first test of subject 'Data Structures and Algorithms' for N students. Compute

1. The average score of class
2. Highest score and lowest score of class
3. Marks scored by most of the students
4. List of students who were absent for the test

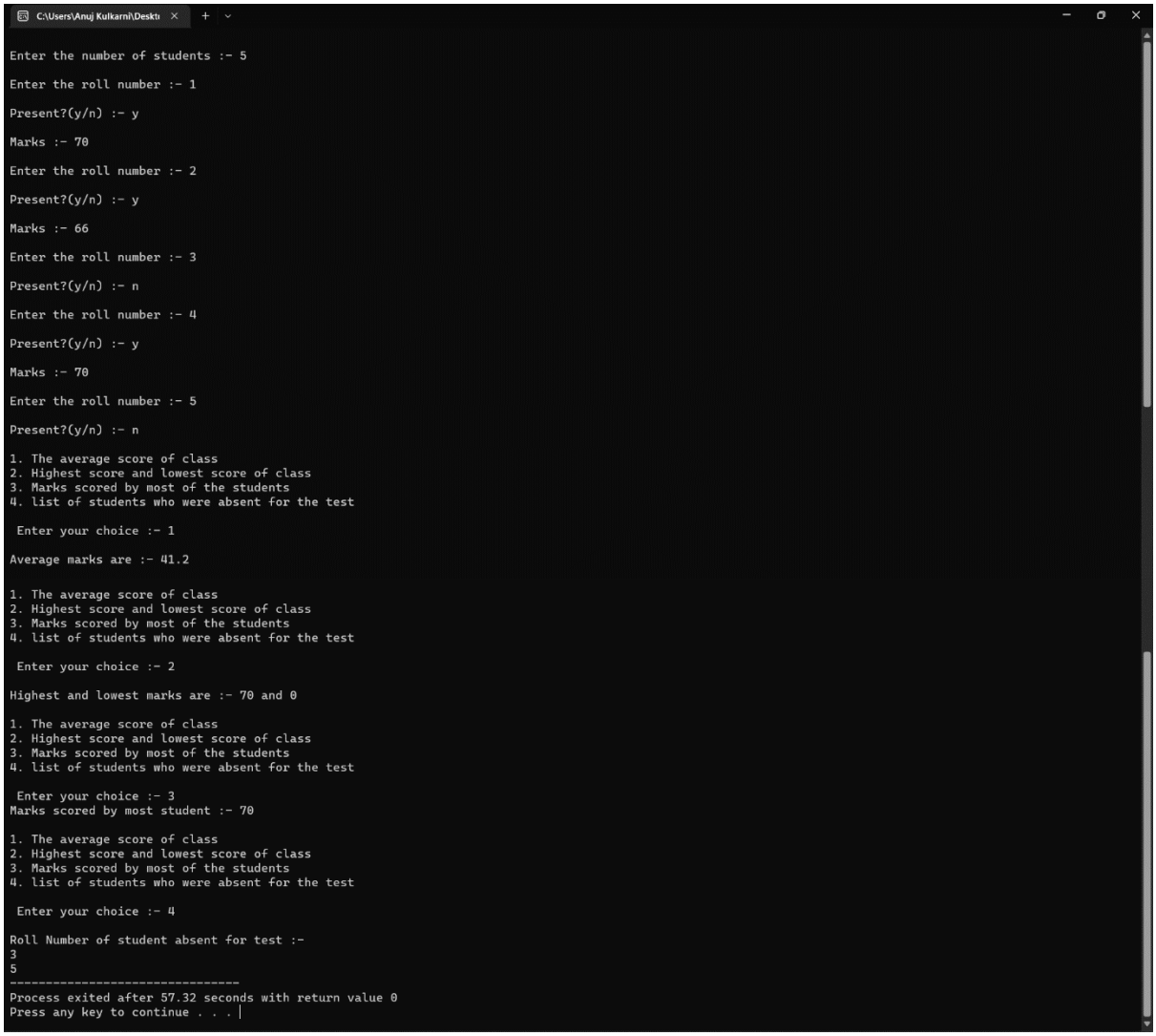
```
#include<iostream>
using namespace std;
class student
{
   int roll;
   bool present;
   int marks;
   public :
   void getdata();
   int  getmarks();
   bool getpresent();
   int getroll();
};
int student::getroll()
{
   return roll;
}
bool student::getpresent()
{
   if(present)
   {
      return false;
   }
   else
   {
      return true;
   }
}
```

```
int student::getmarks()
{
   return marks;
}
void student::getdata()
{
   char p;
   cout<<"\nEnter the roll number :- ";
   cin>>roll;
   cout<<"\nPresent?(y/n) :- ";
   cin>>p;
   if(p=='y'||p=='Y')
   {
      present=true;
      cout<<"\nMarks :- ";
      cin>>marks;
   }
   else
   {
   present=false;
   marks=0;
   }
}
int main()
{
   int n,choice;
   float avg;
   int c=0,previousc=0,marks;
   int high=-1,low=999;
   cout<<"\nEnter the number of students :- ";
   cin>>n;
   student *s=new student[n];
   for(int i=0;i<n;i++)
   {
      s[i].getdata();
   }
```

```
  for(int k=1;k<=4;k++)
  {
     cout<<"\n1. The average score of class\n2. Highest score and lowest score of class\n3. Marks scored by most of the students\n4. list of students who were absent for the test\n\n Enter your choice :- ";
     cin>>choice;
     switch(choice)
     {
        case 1:
        avg=0;
        for(int i=0;i<n;i++)
        {
           avg+=s[i].getmarks();
        }
        avg=avg/n;
        cout<<"\nAverage marks are :- "<<avg<<endl;
        break;
        case 2:
        for(int i=0;i<n;i++)
        {
           if(s[i].getmarks()>high)
           {
              high=s[i].getmarks();
           }
           if(s[i].getmarks()<low)
           {
              low=s[i].getmarks();
           }
        }
        cout<<"\nHighest and lowest marks are :- "<<high<<"and"<<low<<endl;
        break;
        case 3:
        for(int i=0;i<n;i++)
        {
           for(int j=0;j<n;j++)
           {
```

```
            c=0;
            if(s[i].getmarks()== s[j].getmarks())
            {
               c++;
               if(previousc<c)
               {
                  marks=s[i].getmarks();
                  previousc=c;
               }
            }
         }
      }
      cout<<"Marks scored by most student :- "<<marks<<endl;
      break;
      case 4:
      cout<<"\nRoll Number of student absent for test :- ";
      for(int i=0;i<n;i++)
      {
         if(s[i].getpresent())
         {
            cout<<"\n"<<s[i].getroll();
         }
      }
      break;
   }
  }
 return 0;
}
```

OUTPUT :-


C:\Users\Anuj Kulkarni\Deskt
Enter the number of students :- 5
Enter the roll number :- 1
Present?(y/n) :- y
Marks :- 70
Enter the roll number :- 2
Present?(y/n) :- y
Marks :- 66
Enter the roll number :- 3
Present?(y/n) :- n
Enter the roll number :- 4
Present?(y/n) :- y
Marks :- 70
Enter the roll number :- 5
Present?(y/n) :- n
1. The average score of class
2. Highest score and lowest score of class
3. Marks scored by most of the students
4. list of students who were absent for the test
Enter your choice :- 1
Average marks are :- 41.2
1. The average score of class
2. Highest score and lowest score of class
3. Marks scored by most of the students
4. list of students who were absent for the test
Enter your choice :- 2
Highest and lowest marks are :- 70 and 0
1. The average score of class
2. Highest score and lowest score of class
3. Marks scored by most of the students
4. list of students who were absent for the test
Enter your choice :- 3
Marks scored by most student :- 70
1. The average score of class
2. Highest score and lowest score of class
3. Marks scored by most of the students
4. list of students who were absent for the test
Enter your choice :- 4
Roll Number of student absent for test :-
3
5
Process exited after 57.32 seconds with return value 0
Press any key to continue . . .