**Q1:-** An electricity board charges the following rates to domestic users to discourage large consumption of energy.

For the first 100 units: - 60 P per unit

For the next 200 units: -80 P per unit

Beyond 300 units: -90 P per unit

All users are charged a minimum of Rs 50 if the total amount is more than Rs 300 then an additional surcharge of 15% is added. Implement a C++ program to read the names of users and number of units

consumed and display the charges with names.

**PROGRAM:**

#include<iostream>

using namespace std;

class ElectricityBill {

private:

    string name;

    int units;

    float Charges;

public:

    ElectricityBill(string userName, int unitConsumed) {

        name = userName;

        units = unitConsumed;

        Charges = 0.0;

    }

    void calculateCharges() {

        if (units<=100) {

            Charges=units\*0.60;

        }

        else if (units<=300) {

            Charges= (100\*0.60)+(units-100)\*0.80;

        }

        else {

            Charges= (100\*0.60)+(200\*0.80)+(units-300)\*0.90;

        }

        if (Charges<50) {

            Charges=50;

        }

        if (Charges>300) {

            Charges+=Charges\*0.15;

        }

    }

    void displayBill() {

        cout<<"User Name: "<<name<<endl;

        cout<<"Total Charges: Rs "<<Charges<<endl;

    }

};

int main() {

    string userName;

    int unitConsumed;

    cout << "Enter user's name: ";

    getline(cin, userName);

    cout<<"Enter units consumed: ";

    cin>>unitConsumed;

    ElectricityBill bill(userName, unitConsumed);

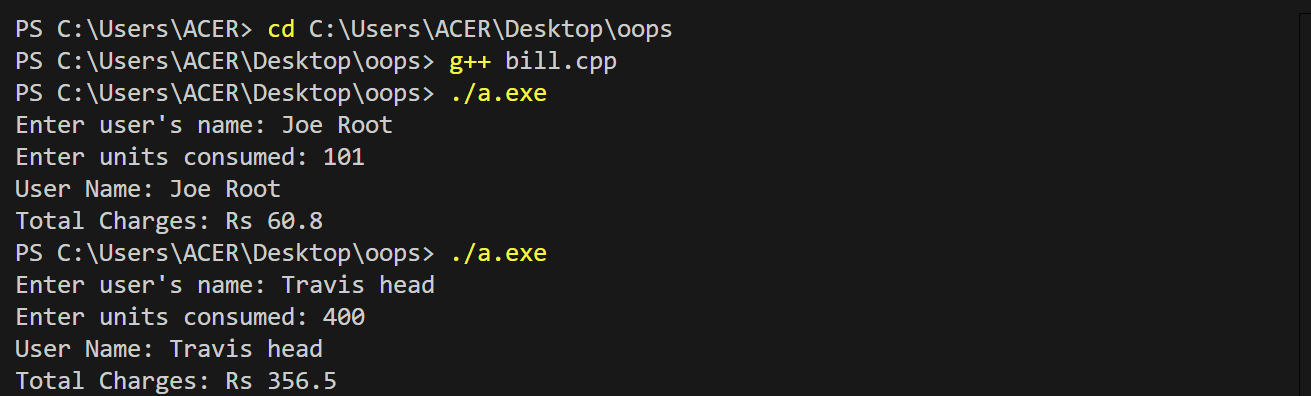
    bill.calculateCharges();

    bill.displayBill();

    return 0;

}

**OUTPUT:**

****

**Q2:-** Construct a C++ program that removes a specific character from a given string and return the updated string.

Typical Input: computer science is the future

Typical Output: compuer science is he fuure

**PROGRAM:**

#include<iostream>

#include<string>

using namespace std;

int main() {

    string str;

    char ch;

    cout<<"Enter the string: ";

    getline(cin,str);

    cout<<"Enter the character to remove: ";

    cin>>ch;

    string newString = "";

    for(int i=0;i<str.length();i++) {

         if (str[i]!=ch) {

            newString+=str[i];

         }

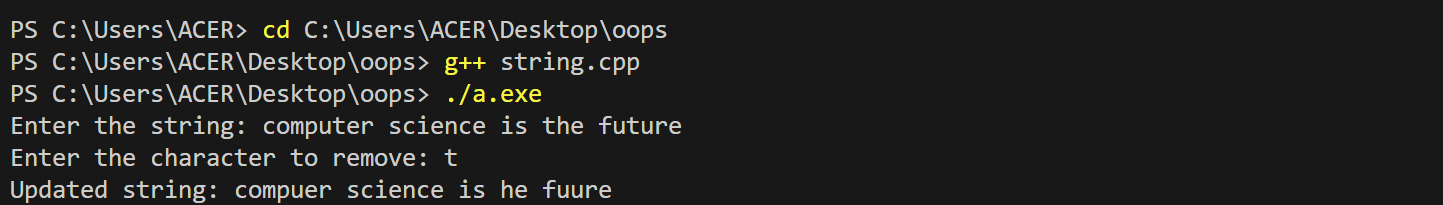
    }

    cout<<"Updated string: "<<newString<<endl;

    return 0;

}

**OUTPUT:**



**Q3:-** Implement a C++ program to find the non-repeating characters in string.

**PROGRAM:**

#include<iostream>

#include<string>

using namespace std;

int main() {

    string str;

    cout<<"Enter a string: ";

    getline(cin, str);

    int Count[256] = {0};

    for(int i=0;i<str.length();i++) {

        if(str[i]!=' ') {

            Count[str[i]]++;

        }

    }

    cout<<"Non-repeating characters are: ";

    for (int i= 0;i<str.length();i++) {

        if (str[i]!=' ' && Count[str[i]] == 1) {

            cout<<str[i];

        }

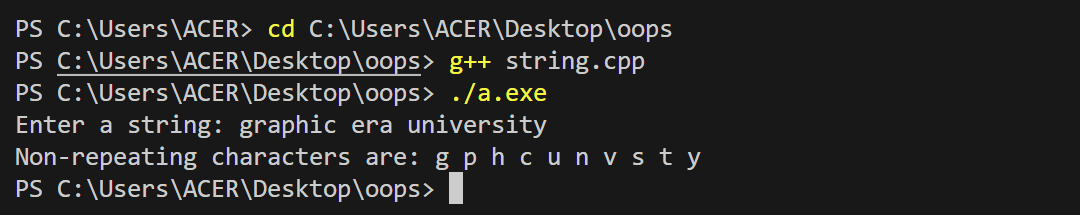
    }

    cout<<endl;

    return 0;

}

**OUTPUT:**



**Q4:-** You are given an array of elements. Now you need to choose the best index of this array. An index of the array is called best if the special sum of this index is maximum across the special sum of all the other indices. To calculate the special sum for any index you pick the first element that is and add it to your sum. Now you pick next two elements i.e., and and add both of them to your sum. Now you will pick the next elements, and this continues till the index for which it is possible to pick the elements. Find the best index and in the output print its corresponding special sum. Note that there may be more than one best index, but you need to only print the maximum special sum.

**PROGRAM:**

#include<iostream>

using namespace std;

int main ()

{

    int n;

    cin>>n;

    int arr[100];

    for(int i=0;i<n;i++)

    {

        cin>>arr[i];

    }

    int sum=-100;

    for(int i= 0;i<n;i++)

    {

        int p=0;//p-> subarray sum

        int index=i;

        int l=1;//l-> length

        while(index+l<=n)

        {

            for(int j=index;j<index+l;j++)

        {

            p+=arr[j];

        }

        index+=l;

        l++;

        }

        if(p>sum)

        {

            sum=p;

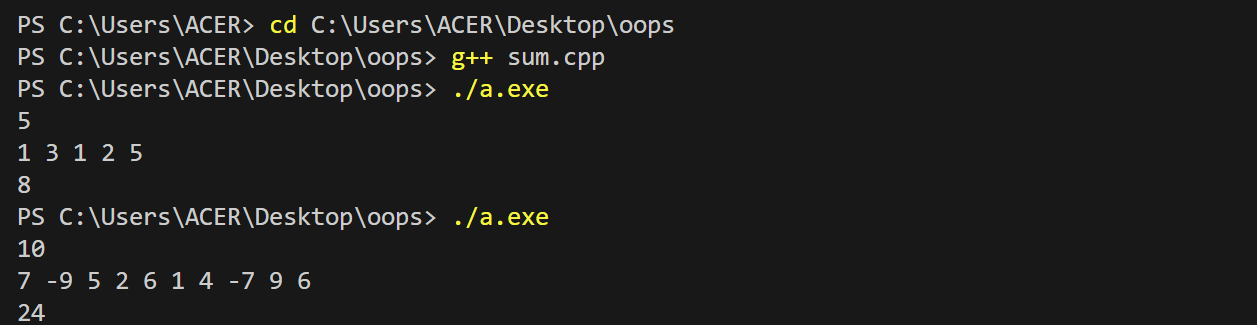
        }

    }

    cout<<sum<<endl;

    return 0;

}

**OUTPUT: **

**Q5**: Implement a C++ program to demonstrate the concept of data abstraction using the concept of Class and Objects.

**PROGRAM:**

#include<iostream>

using namespace std;

class BankAccount {

private:

    float balance;

public:

    BankAccount(float initial\_balance) {

        balance = initial\_balance;

    }

    void deposit(float amount) {

        balance+=amount;

    }

    float getBalance() {

        return balance;

    }

};

int main() {

    float initial\_balance,deposit\_amount;

    cout<<"Enter the initial balance: ";

    cin>>initial\_balance;

    BankAccount myAccount(initial\_balance);

    cout<<"Enter the deposit amount: ";

    cin>>deposit\_amount;

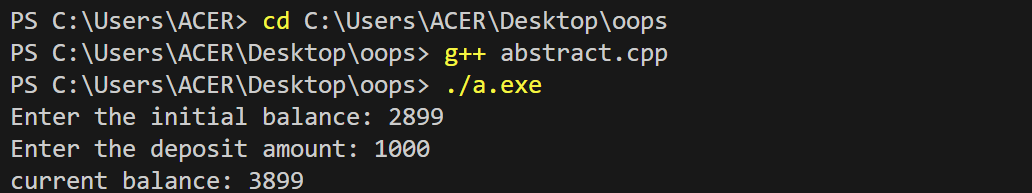
    myAccount.deposit(deposit\_amount);

    cout<<"current balance: "<<myAccount.getBalance()<<endl;

    return 0;

}

**OUTPUT:**



Q6:- Define a class Hotel in C++ with the following specifications

Private members :-

• Rno Data member to store room number

• Name Data member to store customer name

• Tariff Data member to store per day charges

• NOD Data member to store number of days of stay

• CALC() Function to calculate and return amount as NOD\*Tariff ,and if the value of days\* Tariff >10000, then total amount is 1.05\* days\*Tariff.

Public members :-

• Checkin() Function to enter the content Rno, Name, Tariff and NOD

• Checkout() Function to display Rno, Name, Tariff, NOD and Amount (amount to be displayed by calling function) CALC()

**PROGRAM:**

#include<iostream>

#include<string>

using namespace std;

class Hotel {

private:

    int Rno;

    string Name;

    float Tariff;

    int NOD;

    float calc() {

        float amount=NOD\*Tariff;

        if(amount>10000) {

            return 1.05\*amount;

        }

        return amount;

    }

public:

    void Checkin() {

        cout<<"Enter room number: ";

        cin>>Rno;

        cin.ignore();

        cout<<"Enter customer name: ";

        getline(cin, Name);

        cout<<"Enter tariff per day: ";

        cin>>Tariff;

        cout<<"Enter number of days of stay: ";

        cin>>NOD;

    }

    void Checkout() {

        cout<<"Room Number: "<<Rno<<endl;

        cout<<"Customer Name: "<<Name<<endl;

        cout<<"Tariff per day: "<<Tariff<<endl;

        cout<<"Number of days of stay: "<<NOD<<endl;

        cout<<"Amount: "<< calc()<<endl;

    }

};

int main() {

    Hotel guest;

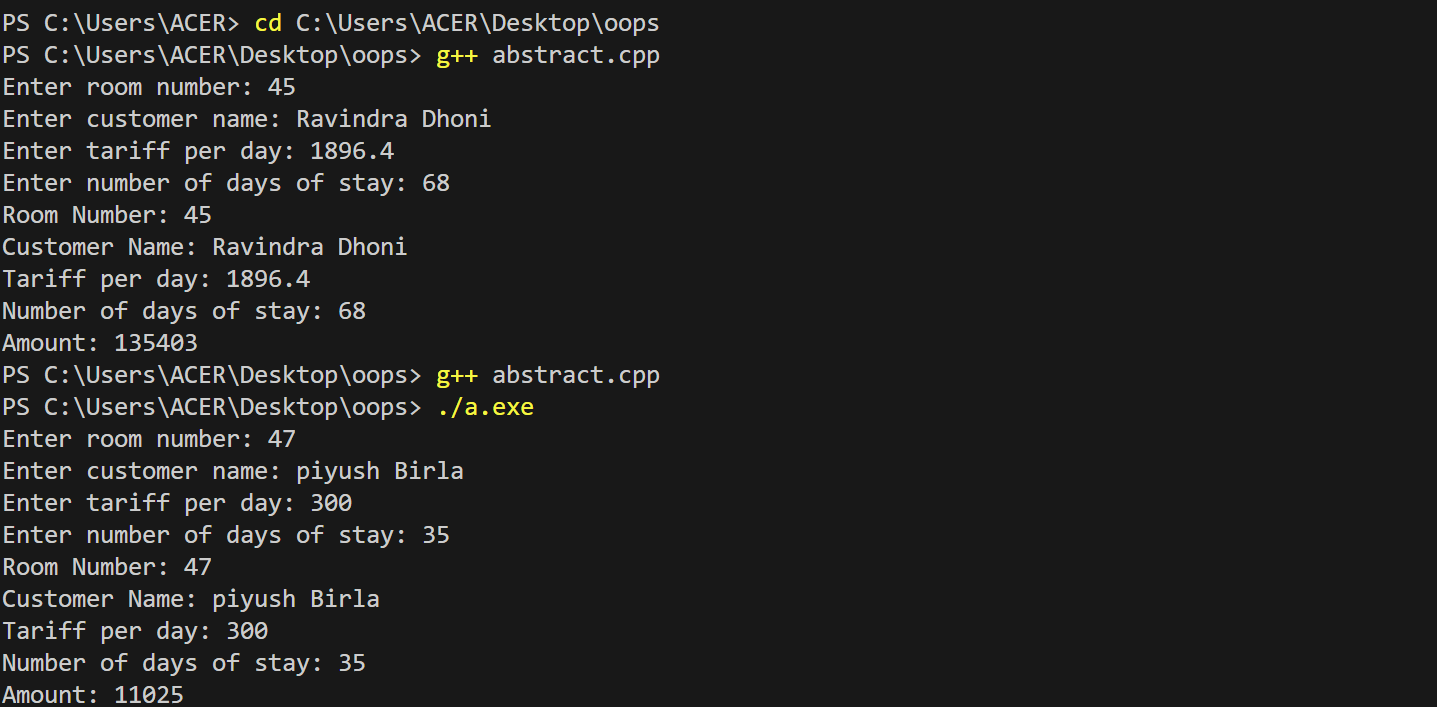
    guest.Checkin();

    guest.Checkout();

    return 0;

}

**OUTPUT:**



**Q8:** Anna is a contender for valedictorian of her high school. She wants to know how many students (if any) have scored higher than her in the exams given during this semester.

Create a class named Student with the following specifications:

1. An instance variable named scores holds a student's 5 exam scores.

2. A void input () function reads 5 integers and saves them to scores.

3. An int calculateTotalScore() function that returns the sum of the student's scores.

**PROGRAM:**

#include<iostream>

using namespace std;

const int max\_std=100;

class Student {

private:

    int scores[5];

public:

    void enter() {

        for(int i=0;i<5;i++) {

            cin>>scores[i];

        }

    }

    int TotalScore(){

        int total=0;

        for(int i=0;i<5;i++) {

            total+=scores[i];

        }

        return total;

    }

};

int main() {

    int n;

    cin>>n;

    if(n>max\_std) {

        cerr<< "Number of students exceeds maximum limit."<<endl;

        return 1;

    }

    Student students[max\_std];

    int totals[max\_std];

    for(int i=0;i<n;i++) {

        students[i].enter();

        totals[i] = students[i].TotalScore();

    }

    int annaTotalScore=totals[0];

    int HigherScores=0;

    for(int i=1;i<n;i++) {

        if(totals[i]>annaTotalScore) {

            HigherScores++;

        }

    }

    cout<<HigherScores<<endl;

    return 0;

}

**OUTPUT:**

