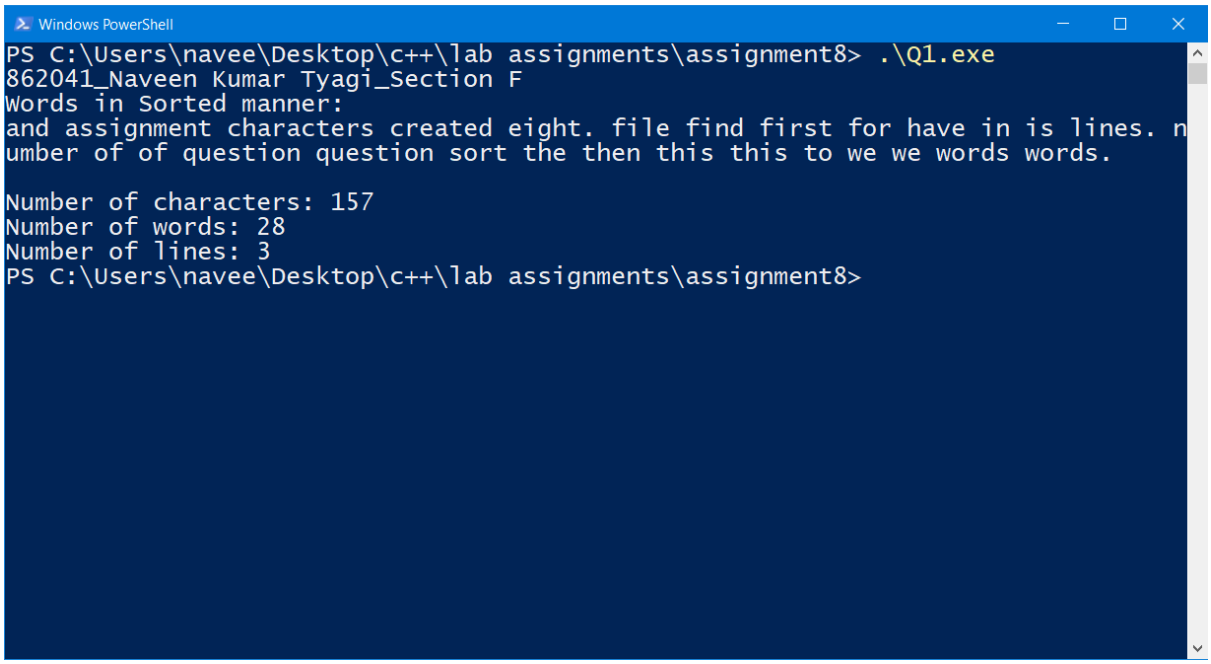
Q1. Write a program to sort the given words and count the number of characters, words, lines in a given input text.

Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
#include<cstring>//for strcmp() and c_str()
#include<fstream>
using namespace std;
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    ifstream file("Q1_text.txt"); //file object instantiated
    file.seekg(0,ios::end); //bring file pointer position to end of file
    int n_characters=file.tellg();
    file.seekg(0,ios::beg); //bring file pointer to begining of file
    string words[100];
    //variable used for index to store words
    //it also represent word count
    int i=0;
    //creating an pointer array
    //pointing to words
    string *ptr_words[100];
    while(!file.eof()){
        file>>words[i]; //storing word to words array
        ptr_words[i]=&words[i]; //storing address
        i++;
    }
    //sorting (insertion sort) of address in pointer array
    //on the basis of names they are pointing
    //below code will sort in lexicographical order
    for(int j=0;j<i-1;j++){
        int loc=j;
        string min=*ptr_words[j];
        for(int k=j;k<i;k++){
            //strcmp is predefined
            //it compares string char by char
            //to do so, it needs character array
            //so c_str() is used
            //it converts string to char array
            int res=strcmp(min.c_str(),(*ptr_words[k]).c_str());
            if(res>0){
                min=*ptr_words[k];
                loc=k;
            }
        }
        //swapping the addresses
        string *temp=ptr_words[j];
        ptr_words[j]=ptr_words[loc];
```

```
        ptr_words[loc]=temp;
    }
    file.seekg(0,ios::beg); //bring position of file pointer to begining of fi
le
    int n_lines=0;//to store number of lines
    char ch;
    while(file>>ch){
        if(ch=='.'){
            n_lines++;
        }
    }
    //print out words in lexicographical order
    cout<<"Words in Sorted manner:\n";
    for(int j=0;j<i;j++){
       cout<<*ptr_words[j]<<" ";
    }
    cout<<endl;
    //print out number of char,words and lines
    cout<<"\nNumber of characters: "<<n_characters;
    cout<<"\nNumber of words: "<<i;  //i represent word count
    cout<<"\nNumber of lines: "<<n_lines;
    file.close();  //file object closed
    return 0;
}
```

Output:



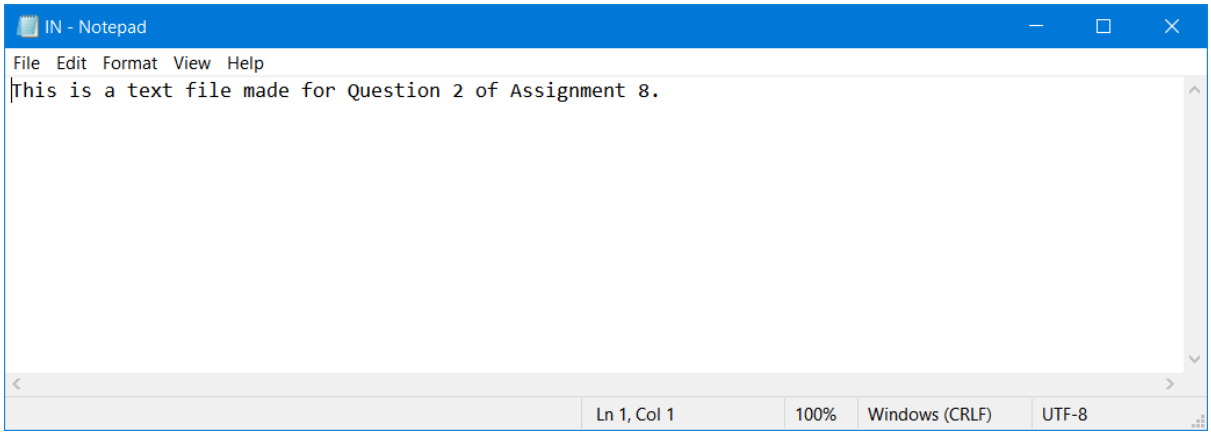
Q2. Write a program to read the content from a text file IN.TXT, count the number of alphabets, digits and special characters present in it and write these information into a text file OUT.TXT .

Code:

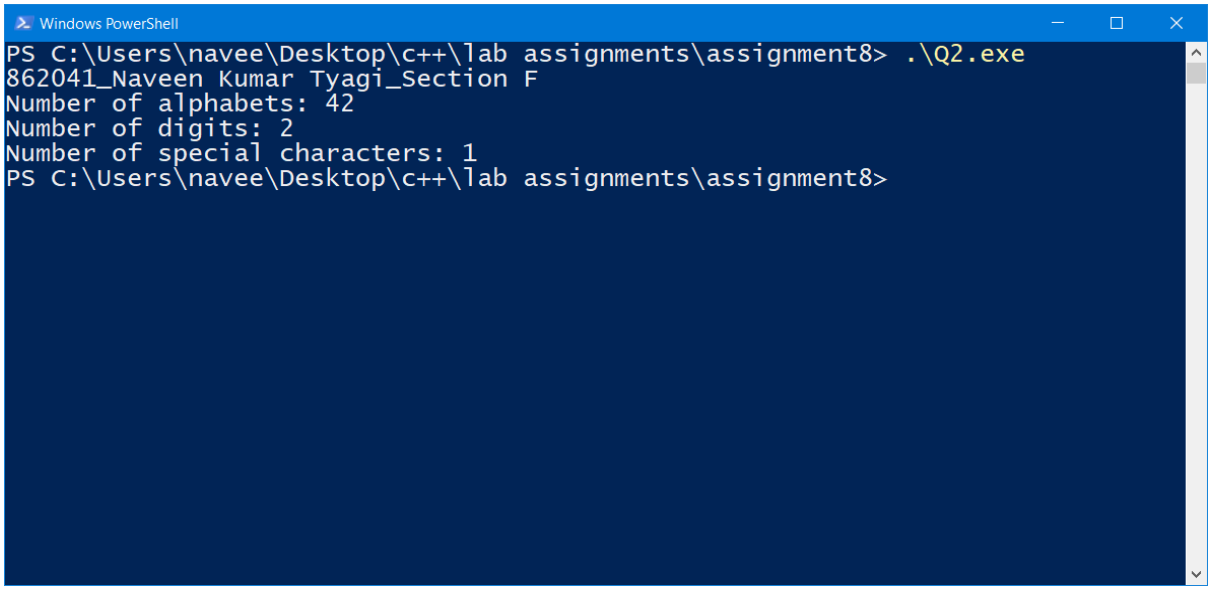
```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
#include<fstream>  //header file for performing operations on files
using namespace std;
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    ifstream file1; //file1 object to read a file
    file1.open("IN.txt");
    ofstream file2; //file2 object to write a file
    file2.open("OUT.txt");

    // variables to store number of alphabets, digits and special character
    int n_alphabets=0,n_digits=0,n_special_char=0;

    //variable to store a character
    //here its purpose to give true value
    //so that while loop continue to run
    //until character get stored in it
    //if we use eof() function in while loop condition
    //then last character will be read by two times
    char ch;
    while(file1>>ch){
        if( ( (ch>=65) && (ch<=90) )|| ( (ch>=97) && (ch<=122) ) ){
            n_alphabets++; //count of alphabets is increased by one
                           //if an alphabet encounter
        }
        else if( (ch>=48) && (ch<=57) ){
            n_digits++;   //count of digit is increased by one
                          //if an numerical digit encounter
        }
        else if( (ch<48) || (ch>57 && ch<65) || (ch>90 && ch<97) || (ch>122)){
            n_special_char++;  //count of special character is increased by on
e
                               // if a special character encounter
        }
    }
    //printing out result in console
    cout<<"Number of alphabets: "<<n_alphabets<<'\n';
    cout<<"Number of digits: "<<n_digits<<'\n';
    cout<<"Number of special characters: "<<n_special_char<<'\n';
    //storing result in file
    file2 <<"Number of alphabets: "<<n_alphabets<<'\n';
    file2 <<"Number of digits: "<<n_digits<<'\n';
    file2 <<"Number of special characters: "<<n_special_char<<'\n';
```

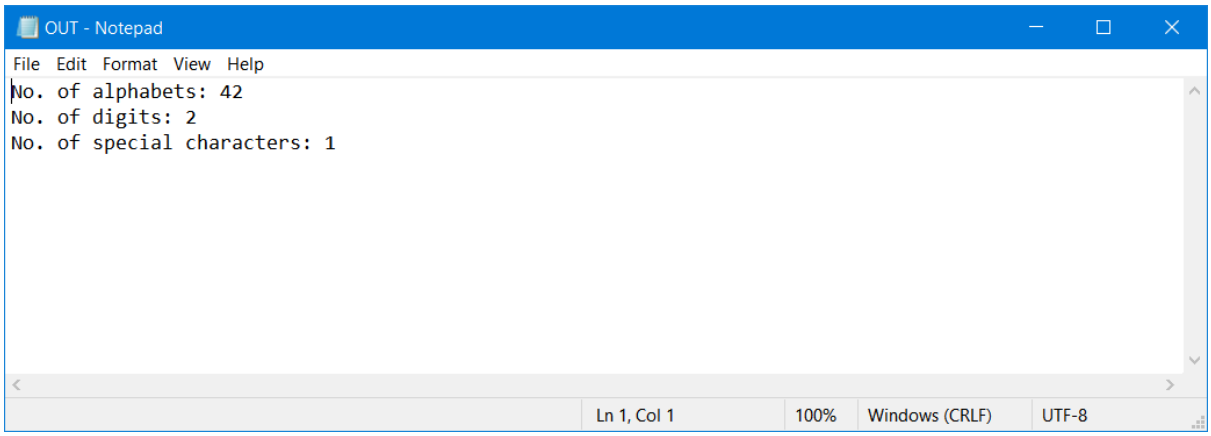
```
    //close file object
    file1.close();
    file2.close();
    return 0;
}
```

Input:



Output:





Q3. Write a function in C++ to count and display the number of lines not starting with ‘A’ present in a text file “ STORY.TXT” Example: If the file “STORY.TXT” contains the following lines,

The rose is red.

A girl is playing there.

There is a playground.

An aeroplane is in the sky.

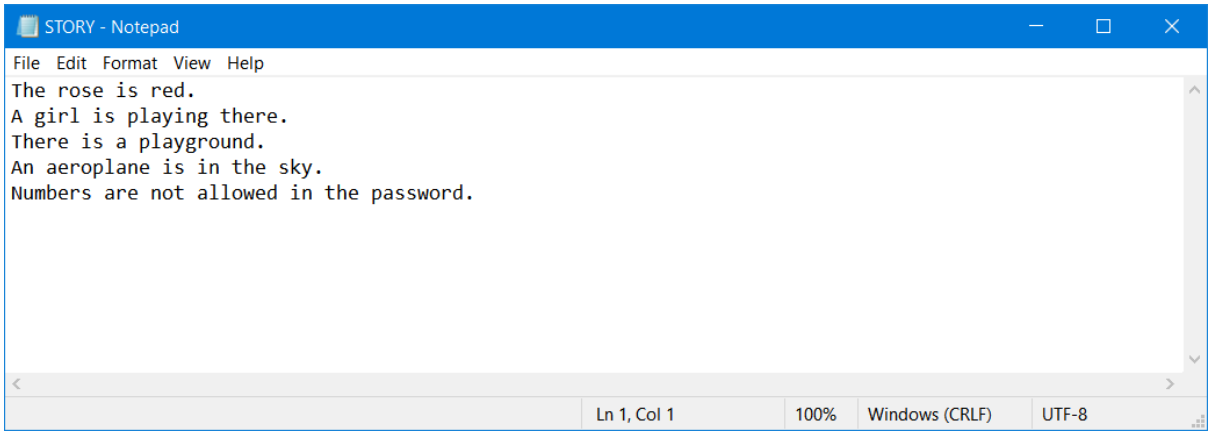
Numbers are not allowed in the password.

The function should display the output as 3

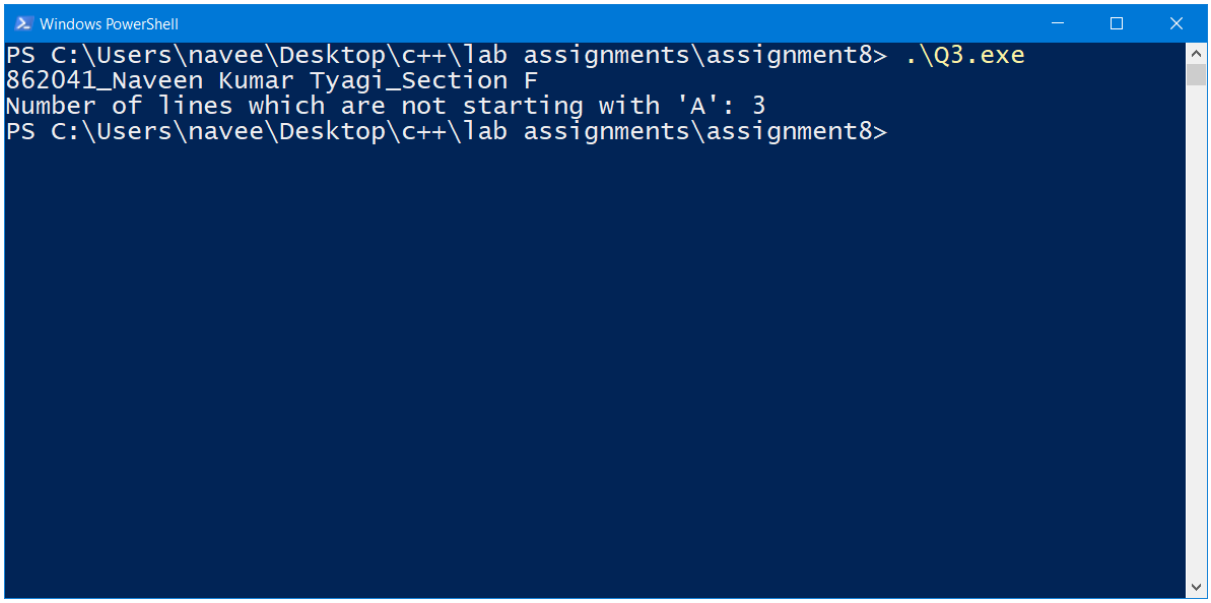
Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
#include<fstream>
using namespace std;
//function to count lines
//that do not start with A
int no_line_notA(ifstream& file){
    int count=0;  //variable to store count
    char line[100];  //array to store line
    while(file.getline(line,100)){
        if(line[0]!='A'){  //counter 'if' statement
            count++;
        }
    }
    return count;  //return count(required no. of lines)
}
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    ifstream file("STORY.txt"); //file object intialization
    //printing out result by calling no_line_notA function
    cout<<"Number of lines which are not starting with \'A\': "<<no_line_notA(
file);
    return 0;
}
```

Input:



Output:



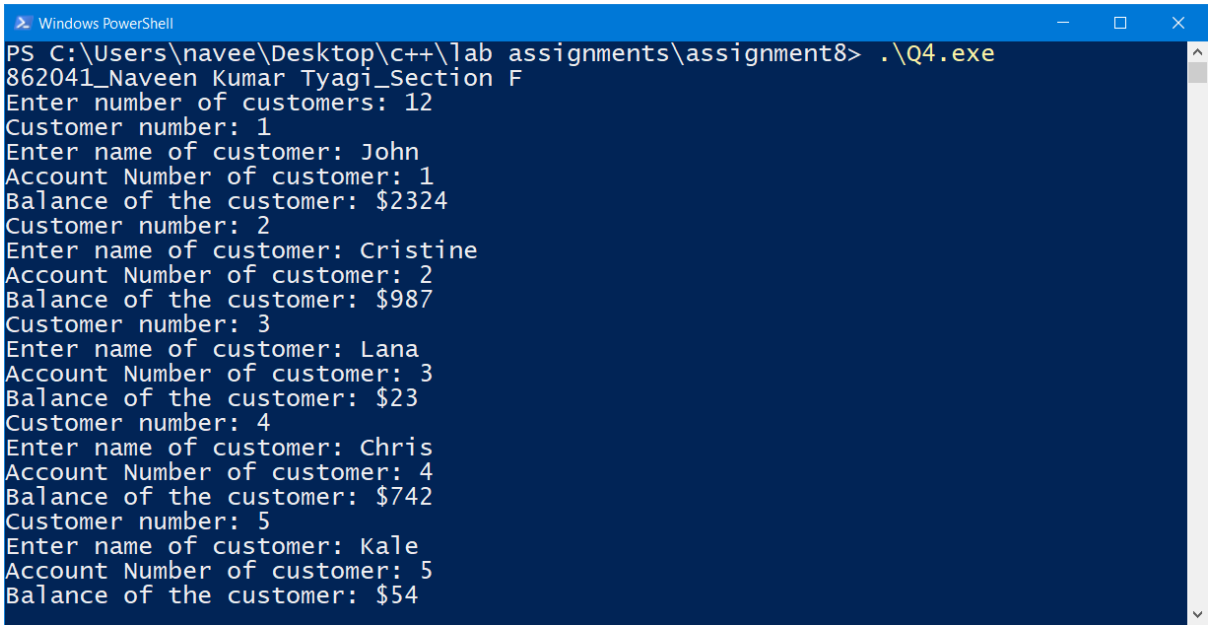
Q4. Write a structure to store the name, account number and balance of customers (more than 10) and store their information. 1 - Write a function to print the names of all the customers having balance less than $200. 2 - Write a function to add $100 in the balance of all the customers having more than $1000 in their balance and then print the incremented value of their balance.

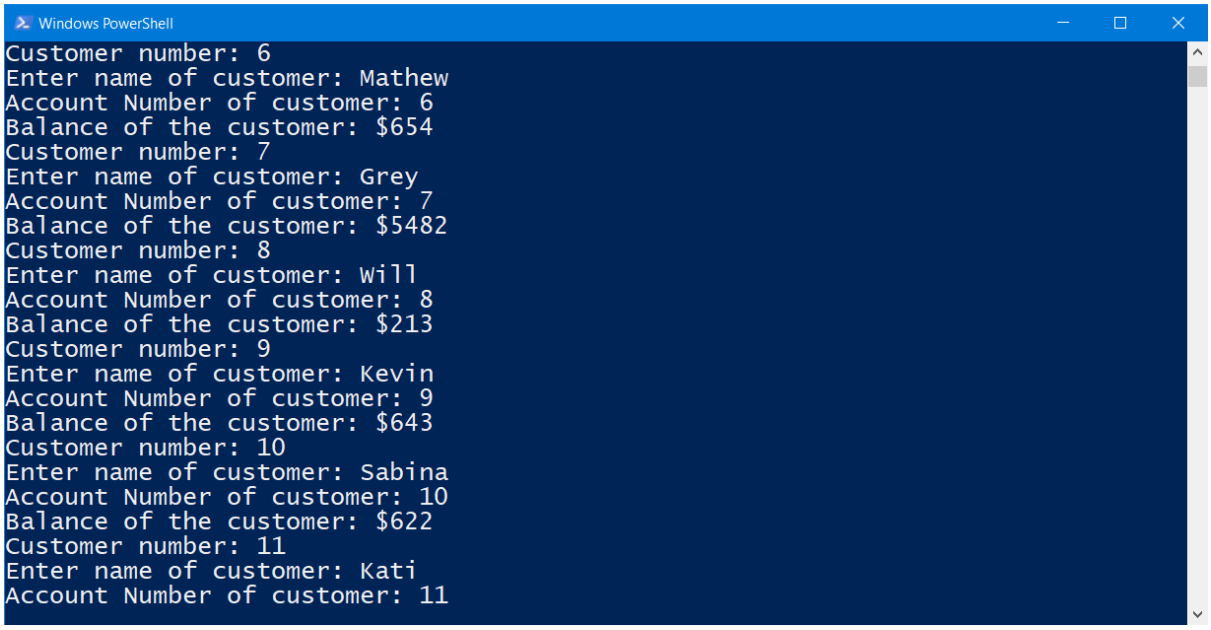
Code:

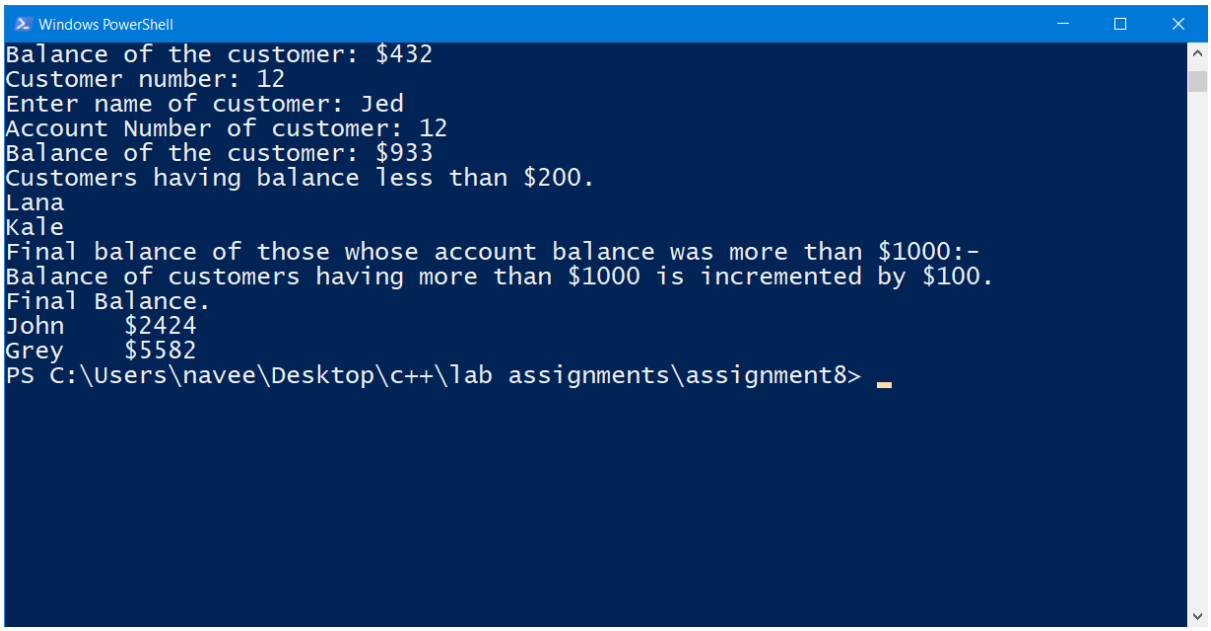
```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
using namespace std;

//required structure
struct BankDetails{
    string name;
    int account_number;
    float balance;
};
//function to print out names of customes whose account balance is less than $
200
void low_balance(struct BankDetails bank_details[],int n_customers){
    cout<<"Customers having balance less than $200.\n";
    for(int i=0; i<n_customers; i++){
        if(bank_details[i].balance<200){
            cout<<bank_details[i].name<<'\n';
        }
    }
}
//functon to increase balance by $100 whose account balance is more than $1000
void increment(struct BankDetails bank_details[],int n_customers){
    cout<<"Balance of customers having more than $1000 is incremented by $100.
\n";
    cout<<"Final Balance.";
    for(int i=0; i<n_customers; i++){
        if(bank_details[i].balance>1000){
            bank_details[i].balance+=100;
            cout<<'\n'<<bank_details[i].name<<"\t$"<<bank_details[i].balance;
        }
    }
}
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    int n_customers; //to store number of customers
    cout<<"Enter number of customers: ";
    cin>>n_customers;
    struct BankDetails bank_details[n_customers];

    //for loop to take customers details from user
    for(int i=0; i<n_customers; i++){
        cout<<"Customer number: "<<i+1<<'\n';
```

```
        cout<<"Enter name of customer: ";
        cin.ignore();
        getline(cin,bank_details[i].name);  // store name of customer
        cout<<"Account Number of customer: ";
        cin>>bank_details[i].account_number; // store account number
        cout<<"Balance of the customer: $";
        cin>>bank_details[i].balance;   //store balance
    }
    //printing out name
    //whose bank balance is less than $200 by low_balance function
    low_balance(bank_details,n_customers);
    //printing out names whose balance is increased
    cout<<"Final balance of those whose account balance was more than $1000:-
\n";
    increment(bank_details,n_customers);
    return 0;
}
```

Output:




Windows PowerShell
Customer number: 6
Enter name of customer: Mathew
Account Number of customer: 6
Balance of the customer: $654
Customer number: 7
Enter name of customer: Grey
Account Number of customer: 7
Balance of the customer: $5482
Customer number: 8
Enter name of customer: Will
Account Number of customer: 8
Balance of the customer: $213
Customer number: 9
Enter name of customer: Kevin
Account Number of customer: 9
Balance of the customer: $643
Customer number: 10
Enter name of customer: Sabina
Account Number of customer: 10
Balance of the customer: $622
Customer number: 11
Enter name of customer: Kati
Account Number of customer: 11


Windows PowerShell
Balance of the customer: $432
Customer number: 12
Enter name of customer: Jed
Account Number of customer: 12
Balance of the customer: $933
Customers having balance less than $200.
Lana
Kale
Final balance of those whose account balance was more than $1000:-
Balance of customers having more than $1000 is incremented by $100.
Final Balance.
John $2424
Grey $5582
PS C:\Users\navee\Desktop\c++\lab assignments\assignment8>

Q5. Write a structure to store the names, salary and hours of work per day of 10 employees in a company. Write a program to increase the salary depending on the number of hours of work per day as follows and then print the name of all the employees along with their final salaries.

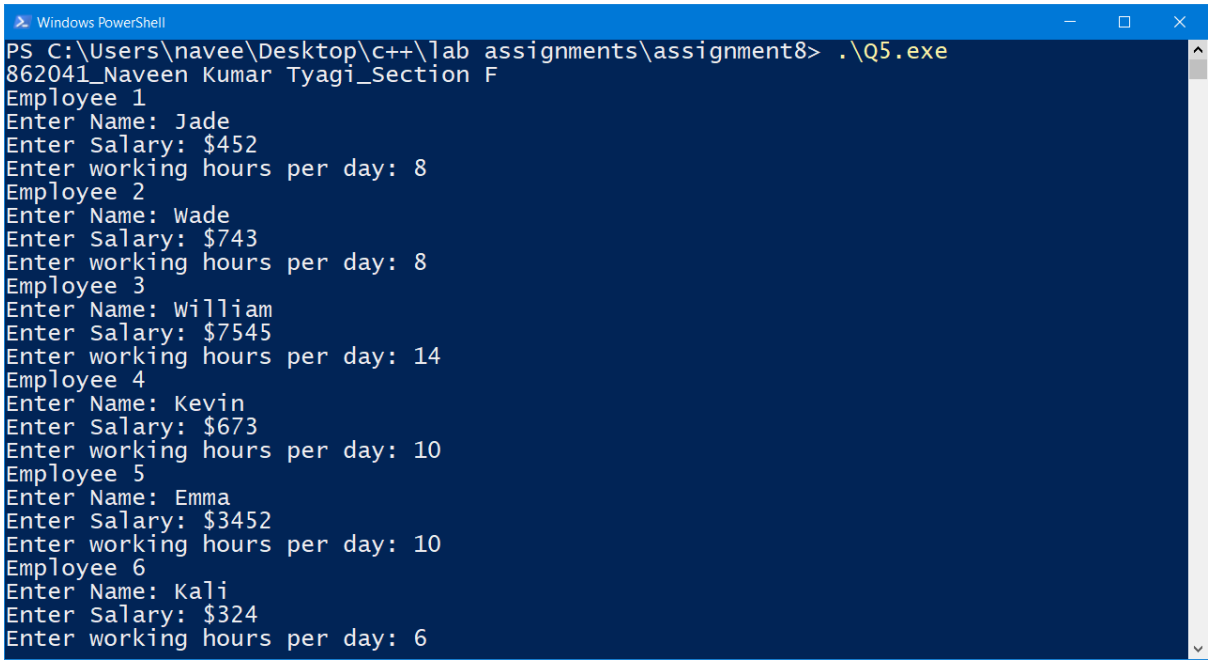
Hours of work per day	8	10	>=12
Increase in salary	$50	$100	$150

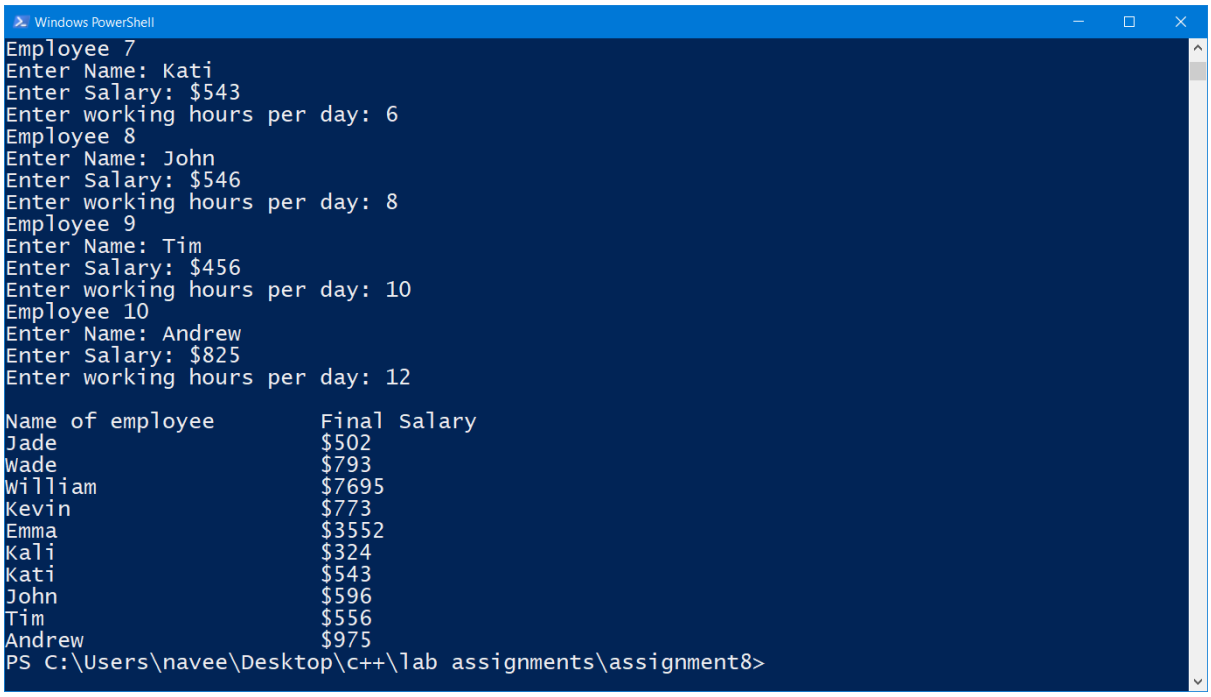
Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
using namespace std;
//required structure
struct Employee{
    string name;
    float salary;
    int hours_per_day;
};
//function to do increment as per the working hours
void increment(struct Employee Employee[]){
    //for loop for performing required operation on each person's salary
    for(int i=0; i<10; i++){
        if(Employee[i].hours_per_day==8){
            Employee[i].salary+=50;
        }
        else if(Employee[i].hours_per_day==10){
            Employee[i].salary+=100;
        }
        else if(Employee[i].hours_per_day>=12){
            Employee[i].salary+=150;
        }
    }
}
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    struct Employee Employee[10]; //structure array
    //for loop to get employee details from user
    for(int i=0; i<10; i++){
        cout<<"Employee "<<i+1<<'\n';
        cout<<"Enter Name: ";
        getline(cin,Employee[i].name);   //for storing name
        cout<<"Enter Salary: $";
        cin>>Employee[i].salary;         //for storing salary
        cout<<"Enter working hours per day: ";
        cin>>Employee[i].hours_per_day;  //for storing hours
        cin.ignore();
    }
    //calling function that perform required increment
```

```
    increment(Employee);
    //for loop to print out names and final salaries of employees
    cout<<"Name of employee\tSalary";
    for(int i=0; i<10; i++){
        cout<<'\n'<<Employee[i].name<<"\t\t$"<<Employee[i].salary;
    }
    return 0;
}
```

Output:





Q6. Write a program to create a structure called Student. The annual examination is conducted for 10 students for three subjects. Write a program to read the data and determine the following:

(a) Total marks obtained by each student.

(b) The highest marks in each subject and the Roll No. of the student who secured it.

(c) The student who obtained the highest total marks.

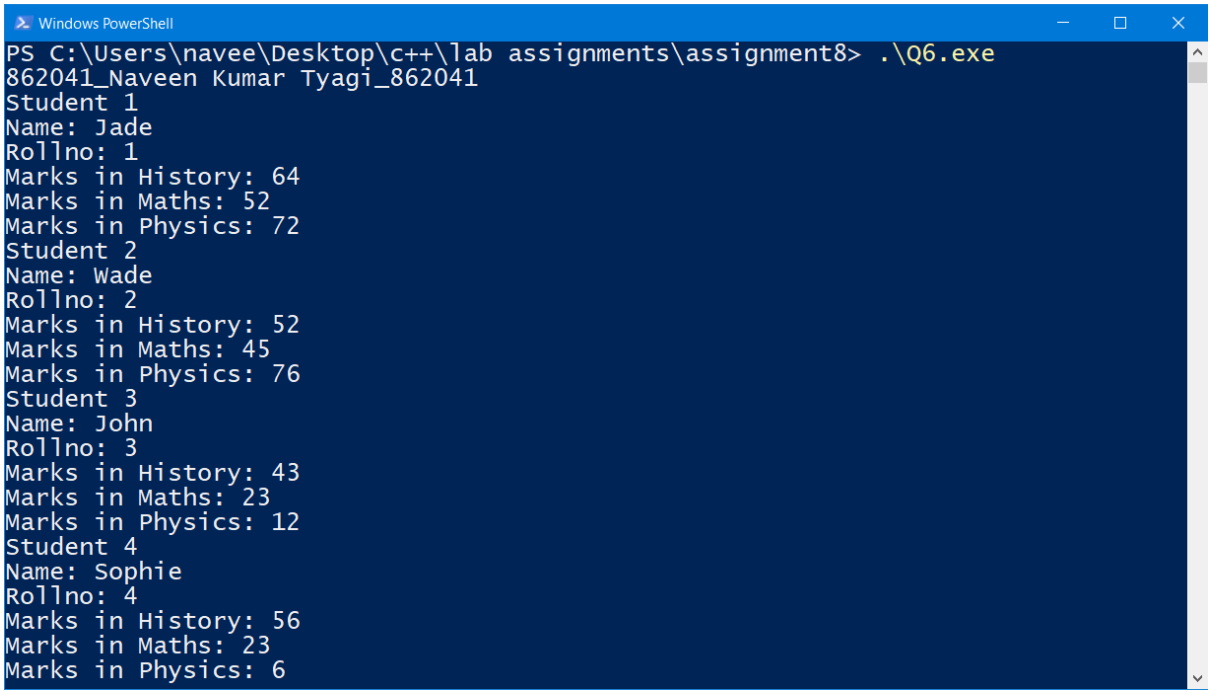
Code:

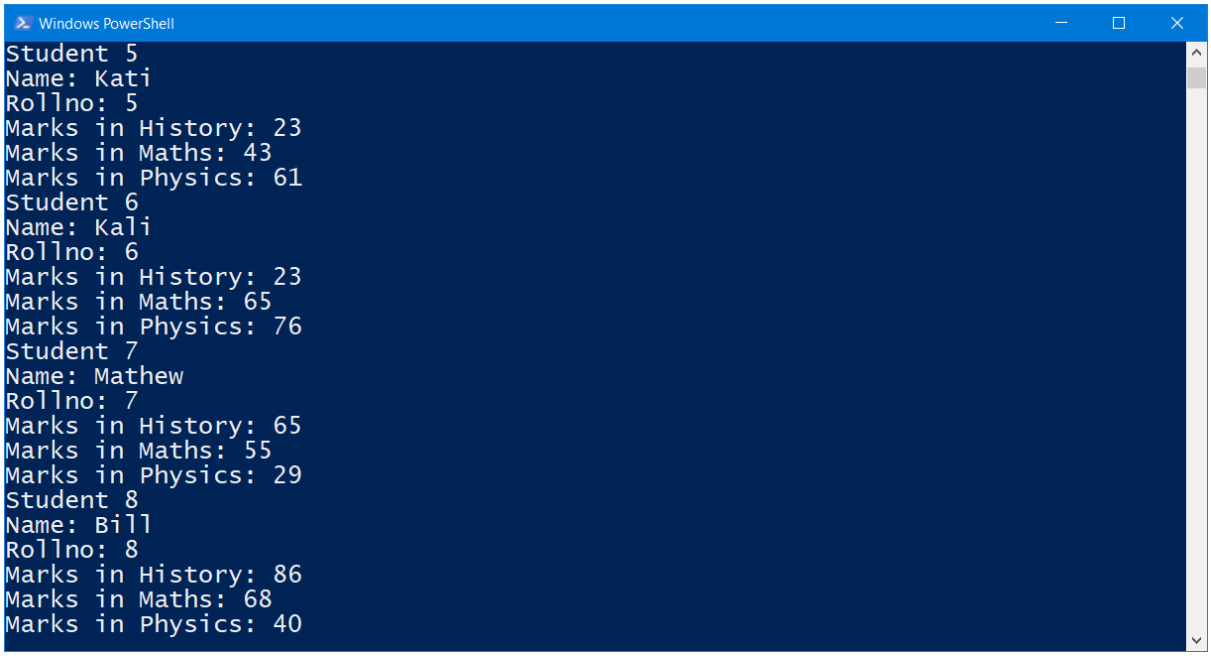
```
//862041_Naveen Kumar Tyagi_862041
#include<iostream>
using namespace std;
//required structure
struct Student{
    string name;
    int rollno;
    int m_history;  //for history marks
    int m_maths;     //for maths marks
    int m_physics;  //for physics marks
    int t_marks;    // for total marks
};
//function to get highest marks in subjects and highest total
void highest_marks(struct Student Student[]){
    //to store highest marks in history and rollno of student who scored
    //assuming first student has scored highest marks
    int highest_history=Student[0].m_history, rollno_history=Student[0].rollno
;

    //to store highest marks in maths and rollno of student who scored
    //assuming first student has scored highest marks
    int highest_maths=Student[0].m_maths, rollno_maths=Student[0].rollno;

    //to store highest marks in physics and rollno of student who scored
    //assuming first student has scored highest marks
    int highest_physics=Student[0].m_physics, rollno_physics=Student[0].rollno
;

    //to store highest total marks and name of student who scored
    //assuming first student has scored highest marks
    int highest_total=Student[0].t_marks;
    string highest_name=Student[0].name;

    //for loop to for comparison and storing highest marks
    for(int i=1; i<10; i++){
        //for history
        if(highest_history<=Student[i].m_history){
            highest_history=Student[i].m_history;
            rollno_history=Student[i].rollno;
        }
```

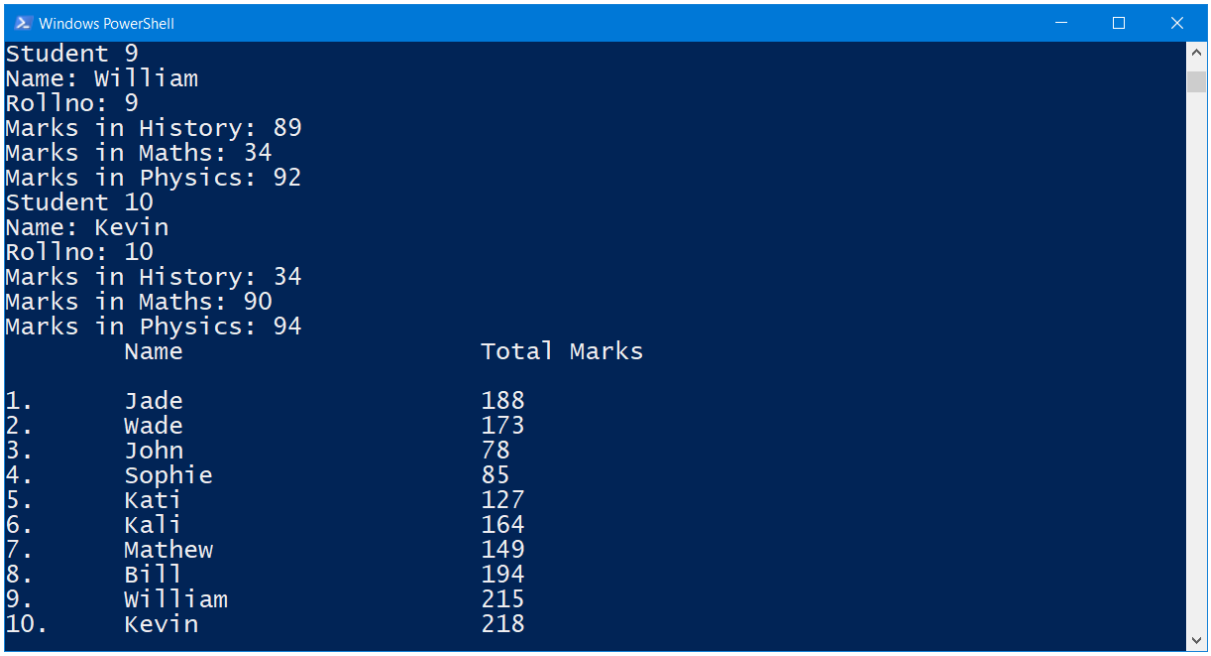
```
        //for maths
        if(highest_maths<=Student[i].m_maths){
            highest_maths=Student[i].m_maths;
            rollno_maths=Student[i].rollno;
        }
        //for physics
        if(highest_physics<=Student[i].m_physics){
            highest_physics=Student[i].m_physics;
            rollno_physics=Student[i].rollno;
        }
        //for total
        if(highest_total<=Student[i].t_marks){
            highest_total=Student[i].t_marks;
            highest_name=Student[i].name;
        }
    }

    //cout statements to print out highest marks
    //in history
    cout<<"\nHighest marks in History: "<<highest_history;
    cout<<"\nRollno of student who secured highest marks in History: "<<rollno
_history<<'\n';
    //in maths
    cout<<"\nHighest marks in Mathamtics: "<<highest_maths;
    cout<<"\nRollno of student who secured highest marks in Mathematics: "<<ro
llno_maths<<'\n';
    //in physics
    cout<<"\nHighest marks in Physics: "<<highest_physics;
    cout<<"\nRollno of student who secured highest marks in Physics: "<<rollno
_physics<<'\n';
    //total
    cout<<"\nHighest total marks: "<<highest_total;
    cout<<"\nRollno of student who secured highest total: "<<highest_name<<'\n
';
}

int main(){
    cout<<"862041_Naveen Kumar Tyagi_862041\n";
    struct Student Student[10];  //structure array
    //for loop to student marks and name from user
    for(int i=0; i<10; i++){
        cout<<"Student "<<i+1<<'\n';
        cout<<"Name: ";
        getline(cin,Student[i].name); //for name
        cout<<"Rollno: ";
        cin>>Student[i].rollno;       //for roll no
        cout<<"Marks in History: ";
        cin>>Student[i].m_history;    //for history
```

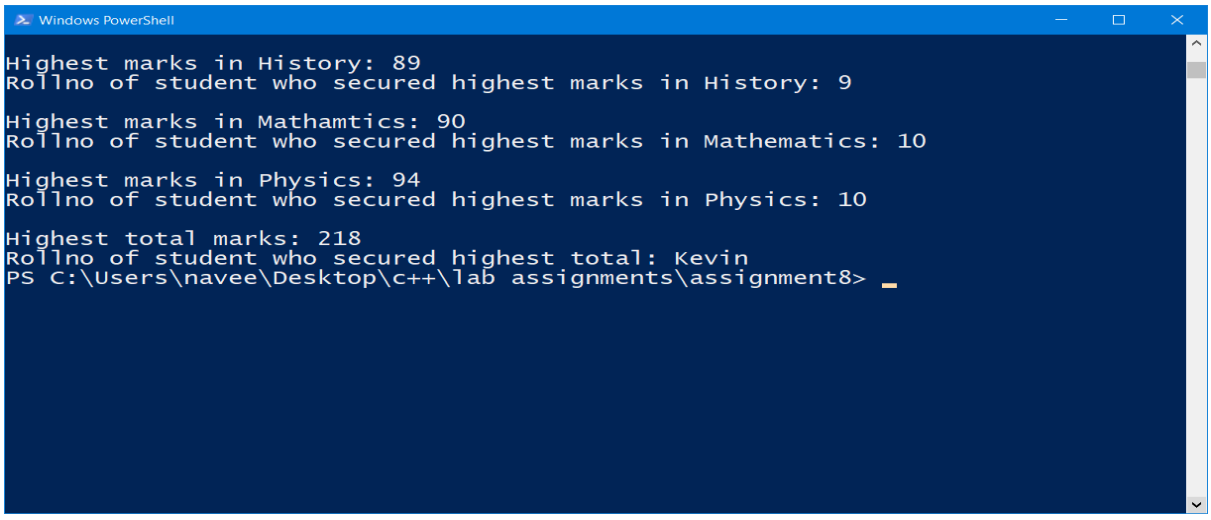
```
        cout<<"Marks in Maths: ";
        cin>>Student[i].m_maths;      //for maths
        cout<<"Marks in Physics: ";
        cin>>Student[i].m_physics;     //for physics
        //for evaluation for total marks
        Student[i].t_marks=Student[i].m_history + Student[i].m_maths + Student
[i].m_physics;
        cin.ignore(); //clear input buffer
    }
    //for loop to print out name and total marks of student s
    cout<<"  \tName\t\t\tTotal Marks\n\n";
    for(int i=0; i<10; i++){
        cout<<i+1<<".\t"<<Student[i].name<<"\t\t\t"<<Student[i].t_marks<<"\n";
    }
    //calling of function to print out highest marks in subject and highest to
tal
    highest_marks(Student);
    return 0;
}
```

Output:




Windows PowerShell
Student 5
Name: Kati
Rollno: 5
Marks in History: 23
Marks in Maths: 43
Marks in Physics: 61
Student 6
Name: Kali
Rollno: 6
Marks in History: 23
Marks in Maths: 65
Marks in Physics: 76
Student 7
Name: Mathew
Rollno: 7
Marks in History: 65
Marks in Maths: 55
Marks in Physics: 29
Student 8
Name: Bill
Rollno: 8
Marks in History: 86
Marks in Maths: 68
Marks in Physics: 40


Windows PowerShell
Student 9
Name: William
Rollno: 9
Marks in History: 89
Marks in Maths: 34
Marks in Physics: 92
Student 10
Name: Kevin
Rollno: 10
Marks in History: 34
Marks in Maths: 90
Marks in Physics: 94
Name Total Marks
1. Jade 188
2. Wade 173
3. John 78
4. Sophie 85
5. Kati 127
6. Kali 164
7. Mathew 149
8. Bill 194
9. William 215
10. Kevin 218

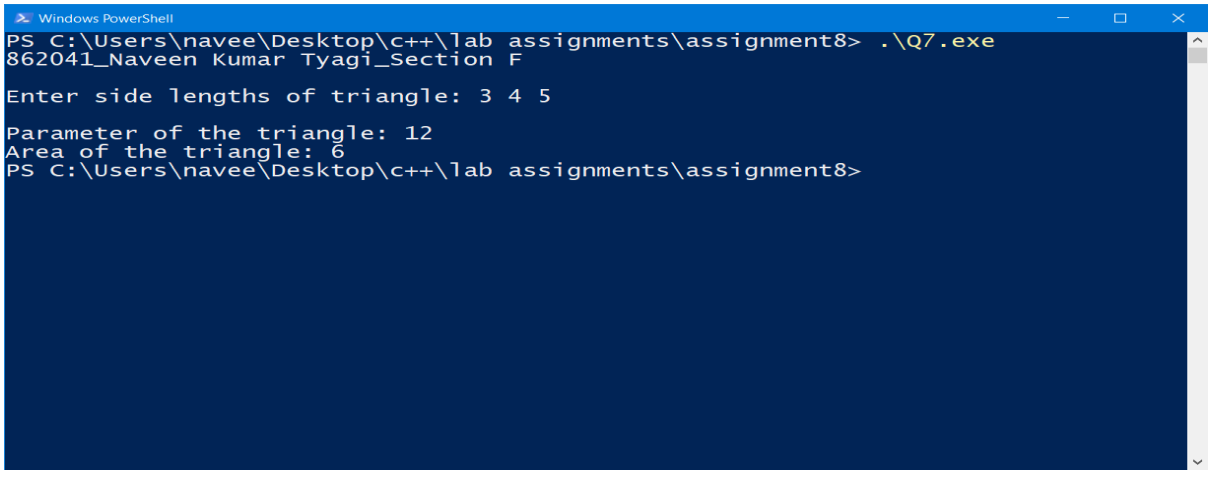

Windows PowerShell
Highest marks in History: 89
Rollno of student who secured highest marks in History: 9
Highest marks in Mathamtics: 90
Rollno of student who secured highest marks in Mathematics: 10
Highest marks in Physics: 94
Rollno of student who secured highest marks in Physics: 10
Highest total marks: 218
Rollno of student who secured highest total: Kevin
PS C:\Users\navee\Desktop\c++\lab assignments\assignment8>

Q7. Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' with the constructor having the three sides as its parameters.

Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
#include<cmath> //for square root function
using namespace std;
//class definition
class Triangle{
    public:
    //constructor to take sides length as parameters
    Triangle(float a,float b,float c){
        float parameter=a+b+c; //to store parameter
        cout<<"\nParameter of the triangle: "<<parameter;  //print perimeter

        float s=parameter/2;  //semi-perimeter
        //calculating area using heron's formula
        float area=sqrt(s*(s-a)*(s-b)*(s-c));
        cout<<"\nArea of the triangle: "<<area; //print area
    }
};
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    //class object
    //parametrized constructor instantiated
    //will print perimeter and area also
    float a,b,c;
    cout<<"\nEnter side lengths of triangle: ";
    cin>>a>>b>>c;
    Triangle A(a,b,c);
    return 0;
}
```

Output:



Q8. Write a program by creating an 'Employee' class having the following functions and print the final salary.

1 - 'getInfo()' which takes the salary, number of hours of work per day of employee as parameters

2 - 'AddSal()' which adds $10 to the salary of the employee if it is less than $500.

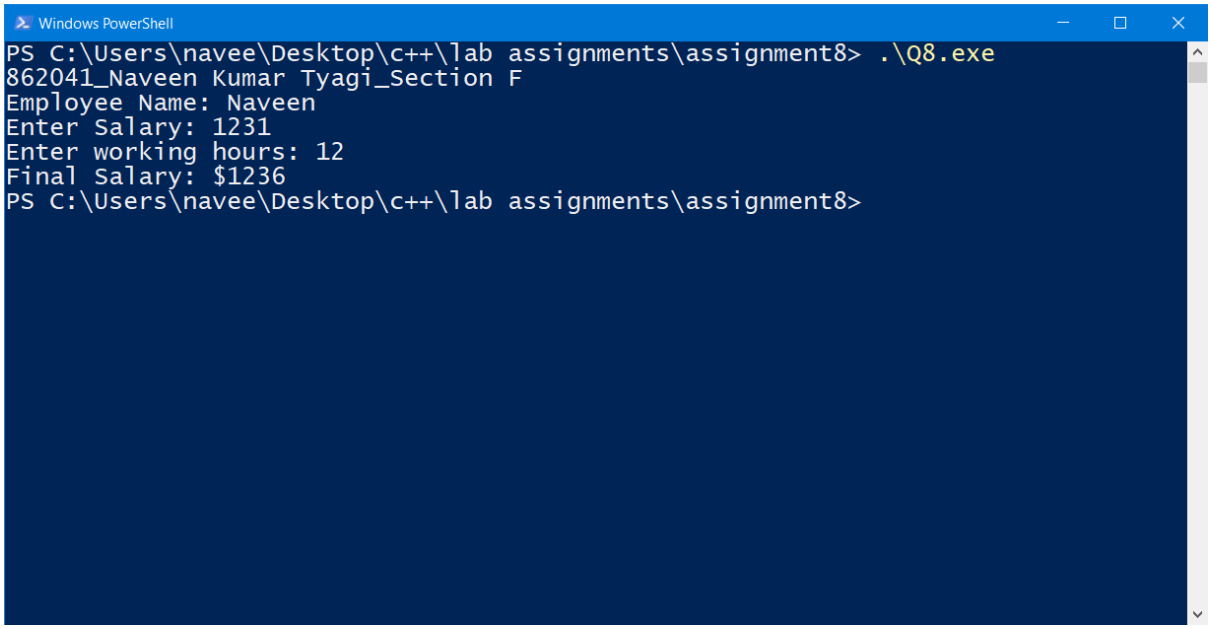
3 - 'AddWork()' which adds $5 to the salary of the employee if the number of hours of work per day is more than 6 hours.

Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
using namespace std;
// required class
class Employee{
    private:
    //function to add $10 to salary
    //this will be called when salary is less than $500
    float AddSal(float salary){
        salary+=10;
        return salary;
    }
    //function to add $5 to salary
    //this will be called when working hours per day is more than 6
    float AddWork(float salary){
        salary+=5;
        return salary;
    }
    public:
    //function which take salary and working hours per day as arguements
    //and call required function to perform desired increment to salary
    //then it will final salary
    void getInfo(float salary,int num_hours){
        if(salary<500){
            salary=AddSal(salary); //function call and storing return value in
 salary
        }
        if(num_hours>6){
            salary=AddWork(salary); //function call and storing return value i
n salary
        }
        cout<<"Final Salary: $"<<salary; //print final salary
    }
};
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    Employee Naveen; //class object instantiated
    float salary;
    int hours;
```

```
    cout<<"Employee Name: Naveen\nEnter Salary: ";
    cin>>salary;
    cout<<"Enter working hours: ";
    cin>>hours;
    Naveen.getInfo(salary,hours); //passing values. this will print final sala
ry also
    return 0;
}
```

Output:

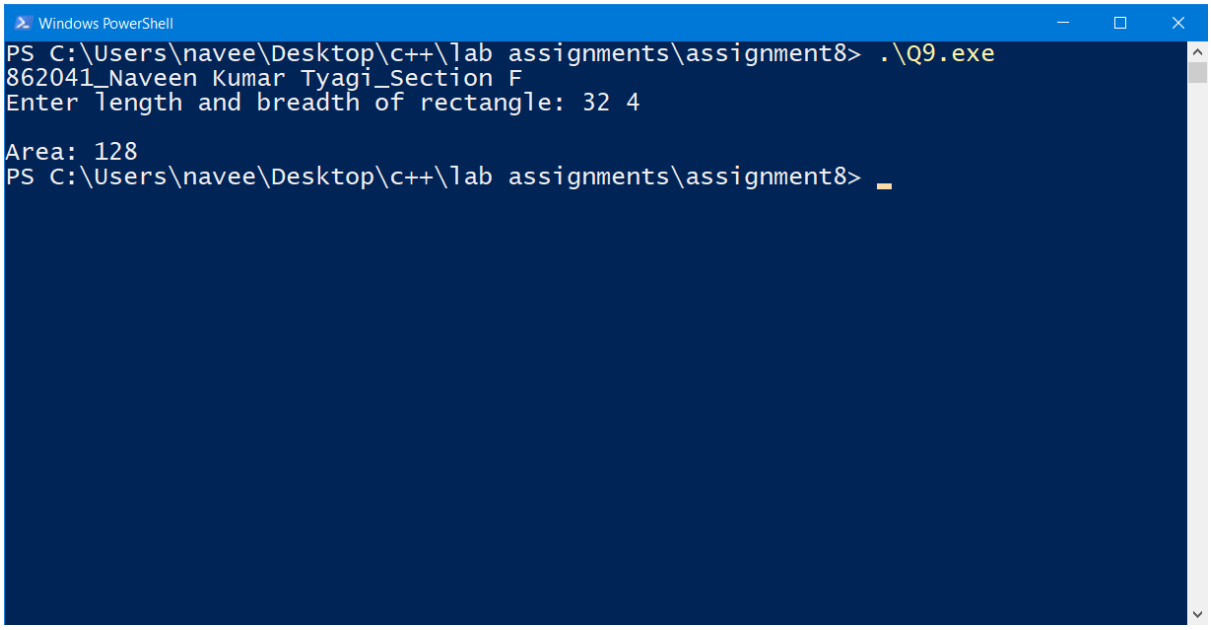


Q9. Write a program to print the area of a rectangle by creating a class named 'Area' having two functions. First function named as 'setDim' takes the length and breadth of the rectangle as parameters and the second function named as 'getArea' returns the area of the rectangle. Length and breadth of the rectangle are entered through keyboard.

Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
using namespace std;
//class for evaluating area of rectangle
class Area{
    private:
    float l,b; //variable to store length and breadth

    //function to evaluate and return area
    float getArea(float l,float b){
        float area=l*b;
        return area;
    }

    public:
    //function to take length and breadth as parameter
    //those will be stored in l an b respectively
    //it also print area finally
    void setDim(float length,float breadth){
        l=length;
        b=breadth;
        cout<<getArea(l,b);//print area
    }
};
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    Area rectangle; //class object instantiated
    float length,breadth;
    cout<<"Enter length and breadth of rectangle: ";
    cin>>length>>breadth; //taking input for length and breadth
    cout<<"\nArea: ";
    rectangle.setDim(length,breadth);  //passing dimension of rectangle and pr
inting area
    return 0;
}
```

Output:


Windows PowerShell
PS C:\Users\navee\Desktop\c++\lab assignments\assignment8> .\Q9.exe
862041_Naveen Kumar Tyagi_Section F
Enter length and breadth of rectangle: 32 4
Area: 128
PS C:\Users\navee\Desktop\c++\lab assignments\assignment8>

Q10. Write the definition for a class called complex that has floating point data members for storing real and imaginary parts. The class has the following member functions: void set(float, float) to set the specified value in object void disp() to display complex number object complex sum(complex) to sum two complex numbers & return complex number

i. Write the definitions for each of the above member functions.
ii. ii. Write main function to create three complex number objects. Set the value in two objects and call sum() to calculate sum and assign it in third object. Display all complex numbers.

Code:

```
//862041_Naveen Kumar Tyagi_Section F
#include<iostream>
#include<cmath>
using namespace std;
//class for complex number and for their addition
class complex{
    float x;  //for storing real part
    float y;  //for storing imaginary part
    public:
    //function to assign real and imaginary part to x and y
    void set(float real,float imaginary){
        x=real;
        y=imaginary;
    }
    //function to display complex and arguement
    //i represents iota
    void disp(){
        cout<<x<<" + i"<<y<<"\targ(radian) = "<<atan(x/y);
    }
    //function for addition with 'complex' datatype
    complex sum(complex Z){
        complex sum;
        sum.x=x + Z.x; //addtion of real part
        sum.y=y + Z.y; //addtion of imaginary part
        return sum;
    }
};
int main(){
    cout<<"862041_Naveen Kumar Tyagi_Section F\n";
    complex Z1, Z2, Z3; //complex class object instantiated
    float re,img; //to  store real and imaginary part

    //taking input for first complex number
    cout<<"Enter a complex number.\n";
    cout<<"Real part: ";
    cin>>re;    //taking input from user for real part
```

```
    cout<<"Imaginary part: ";
    cin>>img;  //taking input from user for imaginary part
    Z1.set(re,img);

    //taking input for second complex number
    cout<<"Enter another complex number.\n";
    cout<<"Real part: ";
    cin>>re;  //taking input from user for real part
    cout<<"Imaginary part: ";
    cin>>img;  //taking input from user for imaginary part
    Z2.set(re,img);

    Z3=Z1.sum(Z2); //calling sum function
    //displaying(printing) the three complex number in console
    cout<<"\nFirst complex number: ";
    Z1.disp();
    cout<<"\nSecond complex number: ";
    Z2.disp();
    cout<<"\nSum of those complex numbers: ";
    Z3.disp();
    return 0;
}
```

Output: