

Practice Questions 02

Question 01:

Write a C++ code that needs to ask the user for her or his email address in the format `firstname.lastname@bahria.edu.pk` OR `firstname.lastname@gmail.com`. The application takes as input this email address, parses the email and replies to the user with first name, last name and host name. A sample run is given below for your convenience. User input is shown in bold. Use `substring()`

Output Example:

Please enter your email address (`firstname.lastname@bahria.edu.pk`):

`khalid.amin@bahria.edu.pk`

First Name: Khalid

Last Name: Amin

Host Name: bahria.edu.pk

Code:

```
#include<iostream>
#include<string>
using namespace std;
int main(){
    string email;
    cout << "Enter your email address (fistname.lastname@abc.com): ";
    getline(cin, email);
    size_t p1 = email.find('.');
    size_t p2 = email.find('@', p1 + 1);

    cout << endl;
    cout << "First Name: " << email.substr(0, p1) << endl;
    cout << "Last Name: " << email.substr(p1 + 1, (p2 - p1) - 1) << endl;
    cout << "Host Name: " << email.substr(p2+1) << endl;
}
```

Output:

```

PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q01.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter your email address (fistname.lastname@abc.com): muhammad.hasann845@gmail.com

First Name: muhammad
Last Name: hasann845
Host Name: gmail.com
PS D:\Hasan\C++\00. University\Practice Questions 02> █

```

Question 02:

Write a program that calculates the user's body mass index (BMI) and categorizes it as underweight, normal, overweight, or obese, based on the table from the United States Centers for Disease Control:

BMI	Weight Status
Below 18.5	Underweight
18.5 – 24.9	Normal
25.0-29.9	Overweight
30.0 and above	Obese

To calculate BMI based on weight in pounds (lb) and height in inches (in), use this formula (rounded to tenths):

$$\text{BMI} = \frac{\text{mass(lb)}}{(\text{height(in)})^2} \times 703$$

Prompt the user to enter weight in pounds and height in inches.

Code:

```

#include<iostream>
using namespace std;

int main(){
    float mass = 0.00, height = 0.00, bmi = 0.00;

    cout << "Enter your mass (lb): ";
    cin >> mass;
    cout << "Enter your height (in): ";
    cin >> height;

    bmi = (mass/(height*height))*float(703);
}

```

```

    cout << "\nYour BMI is: " << bmi << endl;

    if(bmi < 18.5) cout << "Underweight." << endl;
    else if(bmi >= 18.5 && bmi <= 24.9) cout << "Normal." << endl;
    else if(bmi >= 25.0 && bmi <= 29.9) cout << "Overweight." << endl;
    else cout << "Obese." << endl;
}

```

Output:

```

PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q02.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter your mass (lb): 99.208
Enter your height (in): 61.811

Your BMI is: 18.2545
Underweight.

```

Question 03:

Write a program to take an array and find the number of occurrences each number had. The output should be something like this:

Number Occurrences

0	1
2	2
3	2
4	1
21	4
29	4
37	4
42	4
50	1

NOTE: Sort the array elements first and then calculate the frequency of each element. Use **single for-loop** to calculate and print the frequency of each element.

Code:

```

#include<iostream>
using namespace std;

void sort(int a[], int size){

```

```
        for(int i = 0; i < size; i++){
            for(int j = 0; j < size-i-1; j++){
                if(a[j] > a[j+1]){
                    int temp = a[j];
                    a[j] = a[j+1];
                    a[j+1] = temp;
                }
            }
        }
    }
}

int main(){
    int num[10];
    int numWithCalFreq[10]; // what if all numbers are unique, so that's why 10
    int freq[10]; // same reason as above

    // taking elements from user
    for(int i = 0; i < 10; i++){
        cout << "Enter element no. "<<i+1<<": ";
        cin >> num[i];
    }

    // sorting them
    sort(num, 10);

    // calculating frequency
    int uniqueCount = 0;
    for(int i = 0; i < 10; i++){
        bool alreadyInArr = false;

        // first of all checking that if its already present in num with
        // calculated frequency array
        for(int k = 0; k < uniqueCount; k++){
            if(numWithCalFreq[k] == num[i]){
                alreadyInArr = true;
                break;
            }
        }

        if(!alreadyInArr){
            int temp = num[i];
            int freqCount = 1;

            for(int j = 0; j < 10; j++){
                if(i == j) continue;
            }
        }
    }
}
```

```
        if(temp == num[j]) freqCount++;
    }

    numWithCalFreq[uniqueCount] = temp;
    freq[uniqueCount] = freqCount;
    uniqueCount++;
}

// now displaying
cout << endl;
cout << "Number\tFrequency" << endl;
for(int i = 0; i < uniqueCount; i++){
    cout << numWithCalFreq[i] << "\t" << freq[i] << endl;
}
}
```

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q03.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter element no. 1: 6
Enter element no. 2: 3
Enter element no. 3: 6
Enter element no. 4: 4
Enter element no. 5: 6
Enter element no. 6: 2
Enter element no. 7: 8
Enter element no. 8: 8
Enter element no. 9: 4
Enter element no. 10: 1

Number  Frequency
1       1
2       1
3       1
4       2
6       3
8       2
```

Question 04:

Given the following array, display its data graphically by plotting each numeric value as a bar of asterisks (*) as shown in the diagram.

```
int[] array = {10, 19, 5, 1, 7, 14, 0, 7, 5};
```

Element	Value	Histogram
0	10	*****
1	19	*****
2	5	*****
3	1	*
4	7	*****
5	14	*****
6	0	
7	7	*****
8	5	*****

Code:

```
#include<iostream>
#include<vector>
using namespace std;

int main(){
    vector<int> a;
    int size;
    cout << "Enter no. of elements: ";
    cin >> size;

    for(int i = 0; i < size; i++){
        int val;
        cout << "Enter element no. "<<i+1<<": ";
        cin >> val;
        a.push_back(val);
    }

    cout << "\nElement  Value  Histogram" << endl;
    for(int i = 0; i < size; i++){
        cout << i << "          " << a[i] << "          ";
        for(int j = 0; j < a[i]; j++) cout << "*";
        cout << endl;
    }
}
```

}

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q04.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter no. of elements: 6
Enter element no. 1: 3
Enter element no. 2: 2
Enter element no. 3: 9
Enter element no. 4: 1
Enter element no. 5: 0
Enter element no. 6: 2
```

Element	Value	Histogram
0	3	***
1	2	**
2	9	*****
3	1	*
4	0	
5	2	**

Question 05:

Write a program that calculates the total score for students in a class. Suppose the scores are stored in a three-dimensional array named scores. The first index in scores refers to a student, the second refers to an exam, and the third refers to the part of the exam. Suppose there are 7 students, 5 exams, and each exam has two parts--the multiple-choice part and the programming part. So, scores[i][j][0] represents the score on the multiple-choice part for the i's student on the j's exam. Your program displays the total score for each student.

```
{{{7.5, 20.5}, {12, 22.5}, {22, 33.5}, {43, 21.5}, {15, 2.5}},
{{4.5, 21.5}, {12, 22.5}, {12, 34.5}, {12, 20.5}, {14, 9.5}},
{{5.5, 30.5}, {9.4, 2.5}, {13, 33.5}, {11, 23.5}, {16, 2.5}},
{{6.5, 23.5}, {9.4, 32.5}, {13, 34.5}, {11, 20.5}, {16, 7.5}},
{{8.5, 25.5}, {9.4, 52.5}, {13, 36.5}, {13, 24.5}, {16, 2.5}},
{{9.5, 20.5}, {9.4, 42.5}, {13, 31.5}, {12, 20.5}, {16, 6.5}},
{{1.5, 29.5}, {9.4, 22.5}, {19, 30.5}, {10, 30.5}, {19, 5.}}};
```

Code:

```
#include<iostream>
using namespace std;

int main(){
    float scores[3][4][2] =
    {
        { // student 1
            {7.5, 20.5}, // sub 1, two marks
            {12, 22.5},
            {22, 33.5},
            {43, 21.5}
        },
        {
            {4.5, 21.5},
            {13, 23.5},
            {34, 65},
            {45, 54.5}
        },
        {
            {45.5, 90.5},
            {62, 26.5},
            {43, 55},
            {34, 23.4}
        }
    }
```



```
    }  
};  
  
float marks = 0.00;  
  
for(int i = 0; i < 3; i++){  
    cout << "\nStudent no. " << i+1 << endl;  
    for(int j = 0; j < 4; j++){  
        cout << "\tSubject no. " << j+1 << endl;  
        for(int k = 0; k < 2; k++){  
            marks += scores[i][j][k];  
        }  
        cout << "\t\tMarks are: " << marks << endl;  
    }  
}  
}
```

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q05.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
```

Student no. 1

Subject no. 1

Marks are: 28

Subject no. 2

Marks are: 62.5

Subject no. 3

Marks are: 118

Subject no. 4

Marks are: 182.5

Student no. 2

Subject no. 1

Marks are: 208.5

Subject no. 2

Marks are: 245

Subject no. 3

Marks are: 344

Subject no. 4

Marks are: 443.5

Student no. 3

Subject no. 1

Marks are: 579.5

Subject no. 2

Marks are: 668

Subject no. 3

Marks are: 766

Subject no. 4

Marks are: 823.4

Question 06:

Write a C++ application with the following prototypes that returns the user's body mass index (BMI)

double calcluateBMI(double weight, double height)

To calculate BMI based on weight in pounds (lb) and height in inches (in), use this formula:

$$\text{BMI} = \frac{\text{mass(lb)}}{(\text{height(in)})^2} \times 703$$

and

string findStatus(double bmi)

Categorizes it as underweight, normal, overweight, or obese, based on the table from the United States Centers for Disease Control:

BMI	Weight Status
Below 18.5	Underweight
18.5 – 24.9	Normal
25.0-29.9	Overweight
30.0 and above	Obese

Prompt the user to enter weight in pounds and height in inches.

Code:

```
#include<iostream>
using namespace std;

double calBMI(double weight, double height){
    return (weight/(height*height))*double(703);
}

string findStatus(double bmi){
    if(bmi < 18.5) return "Underweight";
    else if(bmi >= 18.5 && bmi <= 24.9) return "Normal";
    else if(bmi >= 25.0 && bmi <= 29.9) return "Overweight";
    else return "Obese";
}

int main(){
    double weight = 0.00, height = 0.00, bmi = 0.00;
    cout << "Enter weight(lb): ";
    cin >> weight;
```

```
    cout << "Enter height(in): ";  
    cin >> height;  
  
    bmi = calBMI(weight, height);  
    cout << "\nYour weight status is: " << findStatus(bmi) << endl;  
}
```

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q06.cpp  
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe  
Enter weight(lb): 78  
Enter height(in): 45  
  
Your weight status is: Overweight
```

Question 07:

Write the following 2 methods:

int ComputeOddSum(int input)

int ComputeEvenSum(int input)

The method **ComputeOddSum** find the sum of all odd numbers less than input.

The method **ComputeEvenSum** find the sum of all even numbers less than input.

Now, test these 2 methods by prompting the user to input a number each time until a negative number is entered.

Code:

```
#include<iostream>
using namespace std;

int computeOddSum(int input){
    int sum = 0;
    for(int i = 0; i < input; i++){
        if(i%2 != 0) sum += i;
    }
    return sum;
}

int computeEvenSum(int input){
    int sum = 0;
    for(int i = 0; i < input; i++) if(i%2 == 0) sum += i;
    return sum;
}

int main(){
    int num = 0;

    while(true){
        cout << "\nEnter a number: ";
        cin >> num;
        if(num < 0){
            cout << "\nNegative number entered! Exiting..." << endl;
            break;
        }
        cout << endl;
        cout << "Sum of all even numbers: " << computeEvenSum(num) << endl;
        cout << "Sum of all odd numbers: " << computeOddSum(num) << endl;
    }
}
```

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q07.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe

Enter a number: 7

Sum of all even numbers: 12
Sum of all odd numbers: 9

Enter a number: 5

Sum of all even numbers: 6
Sum of all odd numbers: 4

Enter a number: 10

Sum of all even numbers: 20
Sum of all odd numbers: 25

Enter a number: 0

Sum of all even numbers: 0
Sum of all odd numbers: 0

Enter a number: -9

Negative number entered! Exiting...
```

Question 08:

Write a recursive method to get sum of all number from 1 up to given number. E.g. Number = 5
Result must be sum (1+2+3+4+5)

Code:

```
#include<iostream>
using namespace std;

int sum(int n){
    if(n == 0) return n;
    return n + sum(n-1);
}

int main(){
    int num = 0;
    cout << "Enter a number: ";
    cin >> num;
    cout << "\nSum is: " << sum(num) << endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q08.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter a number: 6

Sum is: 21
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter a number: 10

Sum is: 55
```

Question 09:

Write a recursive function to compute Nth Fibonacci number. Test and trace for N = 6 is 8.

We remember that a Fibonacci number can be recursively defined as: $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$, where $F_0 = 0$, $F_1 = 1$

Code:

```
#include<iostream>
using namespace std;

int fib(int n){
    if(n == 0 || n == 1) return n;
    return (fib(n-1) + fib(n-2));
}

int main(){
    int num = 0;
    cout << "Enter the no. term in Fibonacci sequence: ";
    cin >> num;

    cout << "\nFibonacci value at "<<num<<" is: " << fib(num - 1) << endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q09.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter the no. term in Fibonacci sequence: 5

Fibonacci value at 5 is: 3
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter the no. term in Fibonacci sequence: 6

Fibonacci value at 6 is: 5
```


Question 10:

Write a recursive function to compute power of a number (X^n). Test and trace for 4^5 ?

Hint: $4^5 = 4 * 4^4$; $4^0 = 1$.

Code:

```
#include<iostream>
using namespace std;

float calPower(float base, float exp){
    if(exp == 0) return 1;
    return base*calPower(base, exp - 1);
}

int main(){
    float base = 0.00, exp = 0.00;
    cout << "Enter base: ";
    cin >> base;
    cout << "Enter exponent: ";
    cin >> exp;

    cout << "\n" << base << " to the power " << exp << " is: " << calPower(base,
exp) << endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q10.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter base: 5
Enter exponent: 5

5 to the power 5 is: 3125
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter base: 3
Enter exponent: 2

3 to the power 2 is: 9
```

Question 11:

Write a recursive method isPalindrome that takes a string and returns true if it is read forwards or backwards. For example,

isPalindrome("mom") → true

isPalindrome("cat") → false

isPalindrome("level") → true

Code:

```
#include<iostream>
#include<string>
using namespace std;

bool isPalindrom(string word, int start, int end){
    if(start >= end) return true; // it means we have successfully go through
    complete array
    if(word[start] != word[end]) return false;
    return isPalindrom(word, start + 1, end - 1);
}

int main(){
    string word;
    cout << "Enter ka word: ";
    cin >> word;

    cout << endl;
    if(isPalindrom(word, 0, word.length()-1)) cout << "Palindrom" << endl;
    else cout << "Not a palindrom" << endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q11.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter ka word: hello

Not a palindrom
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter ka word: mom

Palindrom
```

Question 12:

Implement a structure *Car*, that has the following characteristics:

- a. *brandName*,
- b. *priceNew*, which represents the price of the car when it was new,
- c. *color*, and
- d. *odometer*, which is milo meter shows number of milage travelled by car
- e. A method *getPriceAfterUse()* which should return the price of the car after being used according to the following formula:

$$\text{car price after being used} = \text{priceNew} \times \left(1 - \frac{\text{odometer}}{600,000} \right)$$
- f. A method *updateMilage(double traveledDistance)* that changes the current state of the car by increasing its milage, and
- g. A method *outputDetails()* that will output to the screen all the information of the car, i.e., brand name, price new, price used, color, and odometer.
- h. Create an object of type Car.
- i. Use the method *getPriceAfterUse* on the object created in 'A' then output the result to the screen.
- j. Use the method *updateMilage* on the object created in 'A' by passing a valid value.
- k. Use the method *outputDetails* on the object created in 'A'

Code:

```
#include<iostream>
using namespace std;

struct car{
    string brandName;
    float priceNew = 0.00, odometer = 0.00;

    double updateMilage(double traveledDistance){
        return odometer += traveledDistance;
    }

    float getPriceAfterUse(){
        return priceNew*(1-(odometer/600000));
    }
}
```

```
    }

    void outputDetails(){
        cout << "Brand name: " << brandName << endl;
        cout << "Price new: " << priceNew << endl;
        cout << "Price used: " << getPriceAfterUse() << endl;
    }

};

int main(){
    car c;
    double travelledDist = 0.00;

    cout << "Enter brand name: ";
    getline(cin, c.brandName);
    cout << "Enter original price: ";
    cin >> c.priceNew;
    cout << "Enter reading of odometer: ";
    cin >> c.odometer;

    cout << "\nEnter travelled disatance: ";
    cin >> travelledDist;
    c.updateMilage(travelledDist);

    cout << "\nPrice after use is: " << c.getPriceAfterUse() << endl;

    cout << endl;
    c.outputDetails();
}
```

Output:

```
PS D:\Hasan\C++\00. University\Practice Questions 02> g++ q12.cpp
PS D:\Hasan\C++\00. University\Practice Questions 02> ./a.exe
Enter brand name: Toyota
Enter original price: 900
Enter reading of odometer: 0

Enter travelled disatance: 1000

Price after use is: 898.5

Brand name: Toyota
Price new: 900
Price used: 898.5
```