UNIVERSITY OF VOCATIONAL TECHNOLOGY

100, Kandawala Road, Ratmalana

Bachelor of Technology in Software Technology

Computer Programming

1st Semester

Assignment 02

M.N.M. Nifraz

SOF/21/B2/21

UNIVERSITY OF VOCATIONAL TECHNOLOGY

SRI LANKA

Assignment Template & Feedback Form

Course Title	Bachelor of Technology in Software Technology	Module	IT104021
Student's Name	M.N.M. Nifraz	Batch No.	B2
Reg. No.	SOF/21/B2/21	Semester	01
Resource Person	Mr. Nishantha Anuruddha	Assignment No.	02
Issued on	24/07/2022		
Submitted on	12/08/2022	Received By	

Feedback from the Resource Person:

Key Strength:
To Improve Report:
General Comments:

Grade:

Date: .....

....

Resource Person's Signature Resource Person's Signature

1. Write a C++ program to perform following task.

   a. Create a 2D integer array of 3 columns & 4 rows and initially, set all values to 0

   b. User is able to enter integer values only and the maximum no. of elements is 12.

      i. But whenever user enters (-1) before filling the entire 2D array then program should stop taking inputs.

      ii. Inform user how many values already inserted in each turn

   c. Once data entry process is over, print all values of the array

   d. Find the maximum & minimum values and print them

      i. (-1) & (0) are not to be consider as minimum values

   e. Calculate the average of only the entered values

Source Code:

```
#include <iostream>
#include <iomanip>
#include <cmath>

using namespace std;

int main()
{
    cout << "Number Grid\n"
         << setfill('=') << setw(12) << "\n";

    const int ROWS = 4;
    const int COLUMNS = 3;

    int intArray[ROWS][COLUMNS] = {0};
    int inputCount = 0;
    int sum = 0;

    cout << "Enter " << (ROWS * COLUMNS) << " numbers to fill a " << COLUMNS <<
"x" << ROWS << " grid.\n"
         << "(-1 to stop)\n\n";

    int inputValue = 0; // temporary variable to store the input
    for (int i = 0; i < ROWS && inputValue != -1; i++)
    {
```

```
        cout << "[Row #" << (i + 1) << "/" << ROWS << "]\n";
        for (int j = 0; j < COLUMNS; j++)
        {
            // read number
            cout << "[Column #" << (j + 1) << "/" << COLUMNS << "] number #" <<
(inputCount + 1) << "/" << (ROWS * COLUMNS) << ": ";
            cin >> inputValue;

            if (inputValue == -1) // check if input is -1
            {
                cout << "You've entered -1. The program will stop taking any
further inputs.\n";
                break; // break the loop. the outer loop will also will break
since the outer loop condition fails.
            }

            intArray[i][j] = inputValue; // store the value
            sum += intArray[i][j];       // add the value to sum
            inputCount++;                // increment the count
        }
    }

    if (inputCount == 0) // check if count is 0
    {
        // print the message
        cout << "You haven't entered any values.\n"
             << "The program will exit now.";
        return 0; // return main
    }

    // set min and max to first element
    int max = intArray[0][0];
    int min = intArray[0][0];

    // print results
    cout << "\n"
         << "Results\n"
         << setfill('=') << setw(8) << "\n";

    // print the grid
    cout << "Grid Content:\n";
    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLUMNS; j++)
        {
```

```
            // print each element
            cout << intArray[i][j] << "\t";

            // check if value is 0
            if (intArray[i][j] == 0)
            {
                continue; // skip min, max check
            }

            // check if value is greater than max
            if (intArray[i][j] > max)
            {
                max = intArray[i][j]; // assign to max
            }

            // check if value is less than min
            if (intArray[i][j] < min)
            {
                min = intArray[i][j]; // assign to min
            }
        }
        cout << "\n";
    }

    //print output
    cout << "You have entered " << inputCount << " values.\n";
    cout << "Maximum: " << max << "\n";
    cout << "Minimum: " << min << "\n";
    cout << "Sum of entered values: " << sum << "\n";
    cout << "Average (" << sum << "/" << inputCount << "): " << (double)sum /
inputCount << "\n";
}
```

Sample Output:

```
Number Grid
===========
Enter 12 numbers to fill a 3x4 grid.
(-1 to stop)

[Row #1/4]
[Column #1/3] number #1/12: 3
[Column #2/3] number #2/12: 65
[Column #3/3] number #3/12: 7
[Row #2/4]
[Column #1/3] number #4/12: 91
[Column #2/3] number #5/12: 7
[Column #3/3] number #6/12: 101
[Row #3/4]
[Column #1/3] number #7/12: 5
[Column #2/3] number #8/12: -1
You've entered -1. The program will stop taking any further inputs.

Results
=======
Grid Content:
3       65      7
91      7       101
5       0       0
0       0       0
You have entered 7 values.
Maximum: 101
Minimum: 3
Sum of entered values: 279
Average (279/7): 39.8571
PS C:\Users\NifraZ\OneDrive\Documents\UoVT\Semester 1\computer-programming\assignment-2>
```

2. Write a program to count number of words in a user entered string.

a. (Hint: use C-Style String to store user input value.)

Source Code:

```
#include <iostream>
#include <iomanip>
#include <cmath>

using namespace std;

int main()
{
    cout << "Word Counter\n"
         << setfill('=') << setw(13) << "\n";

    const int MAX_CHARS = 1024;

    //read the text
    cout << "Enter the text (" << MAX_CHARS << " characters max): ";
    char s[MAX_CHARS];
    cin.getline(s, MAX_CHARS);

    int wordCount = 0; //set wordCout to 0

    // variable to hold the flag if last char was space
    // initially set to true, to start counting the first word
    bool isInSpace = true;

    for (int i = 0; i < MAX_CHARS; i++)
    {
        if (s[i] == '\0') //check if NUL character
        {
            break; //break the loop
        }

        if (s[i] == ' ') //check if space
        {
            isInSpace = true; //set space flag
        }
        //if not space AND the last character was space
        else if (isInSpace)
        {
            wordCount++; //increment wordCount
```

```
            isInSpace = false; //remove space flag
        }
    }

    // print results
    cout << "\n"
         << "Results\n"
         << setfill('=') << setw(8) << "\n";
    cout << "Your text contains " << wordCount << " words.";
}
```

Sample Output:

```
Word Counter
============
Enter the text (1024 characters max):   sample text  with some extra    spaces

Results
=======
Your text contains 6 words.
PS C:\Users\NifraZ\OneDrive\Documents\UoVT\Semester 1\computer-programming\assignment-2>
```

3. Create a 2D character array (C-Style String type) (5 rows & 10 columns)

a. Store "Yamaha", "Honda", "Benz", "Tata", "Suzuki" strings

b. Print base memory address without using address (&) operator (See image below)

c. Print each row values using a for loop (See image below)

d. Print each element using two for loops, & you must print only not null values (See image below)

Source Code:

```
#include <iostream>
#include <iomanip>
#include <cmath>

using namespace std;

int main()
{
    // assign the array
    char chArray[5][10] = {
        "Yamaha",
        "Honda",
        "Benz",
        "Tata",
        "Suzuki"};

    // print base memory address
    cout << "chArray = " << chArray << "\n";

    // print each row values (words)
    for (int i = 0; i < 5; i++)
    {
        cout << "chArray[" << i << "] = " << chArray[i] << "\n";
    }

    // print each element
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (chArray[i][j] == '\0') // skip if NUL char
            {
                continue;
            }
```

```
            cout << "chArray[" << i << "][" << j << "] = " << chArray[i][j] <<
"\n";
        }
    }
}
```

Sample Output:

```
chArray = 0x61fde0
chArray[0] = Yamaha
chArray[1] = Honda
chArray[2] = Benz
chArray[3] = Tata
chArray[4] = Suzuki
chArray[0][0] = Y
chArray[0][1] = a
chArray[0][2] = m
chArray[0][3] = a
chArray[0][4] = h
chArray[0][5] = a
chArray[1][0] = H
chArray[1][1] = o
chArray[1][2] = n
chArray[1][3] = d
chArray[1][4] = a
chArray[2][0] = B
chArray[2][1] = e
chArray[2][2] = n
chArray[2][3] = z
chArray[3][0] = T
chArray[3][1] = a
chArray[3][2] = t
chArray[3][3] = a
chArray[4][0] = S
chArray[4][1] = u
chArray[4][2] = z
chArray[4][3] = u
chArray[4][4] = k
chArray[4][5] = i
PS C:\Users\NifraZ\OneDrive\Documents\UoVT\Semester 1\computer-programming\assignment-2>
```

4. User will enter long text with colon separated. example Book:Pen:Pencil:Table:Desk

a. Use String object to store user input value (string str)

b. You must print the values one after another, after splitting the string using delimiter (:)

c. All letters should be UPPERCASE

d. Hint:

i. You can use substring () & find () string functions support

ii. You can use while loop support

Source Code:

```
#include <iostream>
#include <iomanip>
#include <cmath>

using namespace std;

int main()
{
    cout << "Word Splitter & Uppercase\n"
         << setfill('=') << setw(26) << "\n";

    //read text
    cout << "Enter the text (seperate the words using ':' colon): \n";
    string text;
    cin >> text;

    for (int i = 0; i < text.length(); i++)
    {
        // check if char is between small case range
        if (text[i] >= 'a' && text[i] <= 'z')
        {
            text[i] -= 32; // deduct 32 to get the uppercase char
        }

        if (text[i] == ':') // check if char is colon
        {
            text[i] = '\n'; // replace with newline character (to split)
        }
    }

    // print results
```

```
    cout << "\n"
         << "Results\n"
         << setfill('=') << setw(8) << "\n";
    cout << text;
}
```

Sample Output:

```
Word Splitter & Uppercase
=========================
Enter the text (seperate the words using ':' colon):
Apple:cat:user123:SoMething:2022

Results
=======
APPLE
CAT
USER123
SOMETHING
2022
PS C:\Users\NifraZ\OneDrive\Documents\UoVT\Semester 1\computer-programming\assignment-2>
```