Lab Manual

Computer Engineering for Session 2019 (Semester Fall-2019)

Programing Fundamentals

This course is a comprehensive introductory course that is intended for students who have no background in computer programming. The aim of this course is to provide students with a firm foundation of “Basics of Programing Fundamentals” and to make them able to design C++ code of algorithms. The aim is to provide foundation of programing and to polish the skills of understanding computational problems in algorithmic way. The programming language used in this programming course is C++.

Instructor: Ms. Sahar Waqar
Graduate Assistant: Usman Ghani, Muhammad Nazam Maqbool

Lab Manual

Computer Engineering for Session 2019 (Semester Fall-2019)

Programing Fundamentals

(Lab 9)

Target: Function.

i) Guidelines/Instructions:

- Complete lab tasks in lab.
- Must write indented and properly formatted code.
- Design should be modular. All aspects will be evaluated.

Reading Material:

- Book: C++ Programming, from problem analysis to problem design.
- Functions in C++ with examples:
  (https://beginnersbook.com/2017/08/cpp-functions/)
  (http://www.cplusplus.com/doc/tutorial/functions/)
- C++ Functions, Function parameters and function overloading:
  (https://www.w3schools.com/cpp/cpp_functions.asp)
- C++ programming examples:
  (https://www.programiz.com/cpp-programming/examples)

ii) Tasks to be completed in lab:

1. Write a program that will display your name on screen using function with **void** return type. This function will be called from main function. Definition of the function will look like this:

   ```
   void displayName()
   {
       //Your code
   }
   int main()
   {
       //call your function
   }
   ```

2. Modify Q1 and add an input parameter in the function named displayName(string name). This function should display the name passed as input. Write another function named displayNTimes(int n, string name) with return type **void**. This function will call displayName(string name) functions up to n times. In main function, write code for taking n and user’s name as input and will call only displayNTimes function with given parameters and user’s name will be displayed n times. Main function may look like this.

   ```
   int main()
   {
       int n;
       string name;
       cin >> name; cin >> n;
       displayNTimes(n, name);
       return 0;
   ```

```
}
```

If user enters **3** and **Maria** then output of the program is

```
Welcome Maria. Let’s learn functions.
Welcome Maria. Let’s learn functions.
Welcome Maria. Let’s learn functions.
```

3. Write a program that inputs two numbers in main() function, passes these numbers to a function named **findMax** which takes two parameters as inputs and has the return type **int**. The function will return the maximum number. This maximum number should be displayed in main function.
4. Write a program that inputs a number in main() function, passes the number to a function named **calculateFactorial** which takes an integer as input and **finds the factorial of the given input**. Think of the return type of this function. Main function will display the factorial of that number.
5. Write a function named **isEven** which takes an input parameter **n** and returns a bool value i.e. true if number is even and false if number is odd. For example:

   ```
   bool isEven(int n)
   {
           //your code
   }
   int main()
   {
           int n;
           cout << “Please enter number: ”
           cin >> n;
           bool myNumber = isEven(n);
           if(myNumber)
                   cout <<endl << “Number is even”;
           else
                   cout << endl << “Number is odd”;
   }
   ```

6. Write a function named **performOperation** which takes three values as input i.e. two integer operands and one char operator (+, -, /, *, M (findMax)). You should take your inputs i.e. two numbers and one operator in main() function, pass them to a function named **performOperation**. The function applies given operation on two numbers based on the operator entered by user using **switch** statement. Return type of **performOperation** is int. For operation **M,** function will return 0 if operand1 > operand2 and 1 otherwise. Use **findMax** function you developed earlier. Function definition will look like this:
   int calculator(**int** operand1, **int** operand2, **char** operation)
7. Write a function to multiply two numbers. The function will define two parameters as default parameter. Main() function can pass any number of parameters 0 or 1 or 2 parameters to function and the function will multiply these number. Finally, function will return calculated value. Also, you will not use simple * multiply operator to multiply two numbers. (You might need instructions from the Lab TA to solve this problem).
8. Write a program that inputs a number in a global variable. The program calls a function that multiplies that values of global variable by 2. The main function then displays the value of global variable.

iii) Tasks to be submitted:

1. **(Fraction calculator)** Write a program that lets the user perform arithmetic operations on fractions. Fractions are of the form a/b, in which a and b are integers and b 6¼ 0. Your program must be menu driven, allowing the user to select the operation **(+, -, *, or /)** and input the numerator and denominator of each fraction. Furthermore, your program must consist of at least the following functions:

   a. Function menu: This function informs the user about the program's purpose, explains how to enter data, and allows the user to select the operation.

   b. Function addFractions: This function takes as input four integers representing the numerators and denominators of two fractions, adds the fractions, and returns the numerator and denominator of the result.

   c. Function subtractFractions: This function takes as input four integers representing the numerators and denominators of two fractions, subtracts the fractions, and returns the numerator and denominator of the result.

   d. Function multiplyFractions: This function takes as input four integers representing the numerators and denominators of two fractions, multiplies the fractions, and returns the numerators and denominators of the result.

   e. Function divideFractions: This function takes as input four integers representing the numerators and denominators of two fractions, divides the fractions, and returns the numerator and denominator of the result.

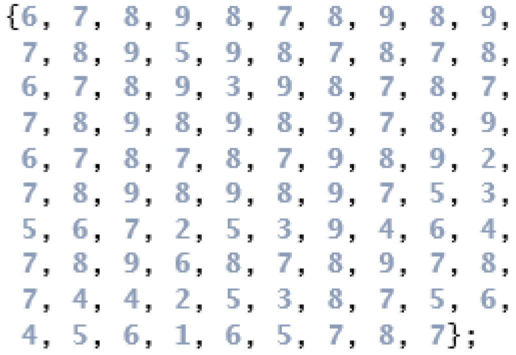
   Some sample outputs are:

```
Welcome to the fraction calculation
Please enter your fractions:
Enter numerator for fraction1 = 3
Enter denominator for fraction1 = 4
Enter numerator for fraction2 = 2
Enter denominator for fraction 2 = 5

Please select your operation:
1.  Press 1 for addition
2.  Press 2 for subtraction
3.  Press 3 for divide
4.  Press 4 for multiply

Enter your operation = 1

3 / 4 + 2 / 5 = 23 / 20

Do you want to continue for same fractions? (Y/N) N
Please enter your fractions:
Enter numerator for fraction1 = 2
Enter denominator for fraction1 = 3
Enter numerator for fraction2 = 3
Enter denominator for fraction 2 = 5

Please select your operation:
1.  Press 1 for addition
2.  Press 2 for subtraction
3.  Press 3 for divide
4.  Press 4 for multiply
```

```
Enter your operation = 4

2 / 3 * 3 / 5 = 6 / 15

Do you want to continue for same fractions? (Y/N) Y

Please select your operation:
1.   Press 1 for addition
2.   Press 2 for subtraction
3.   Press 3 for divide
4.   Press 4 for multiply


Enter your operation = 3

2 / 3 / 3 / 5 = 10 / 9
```

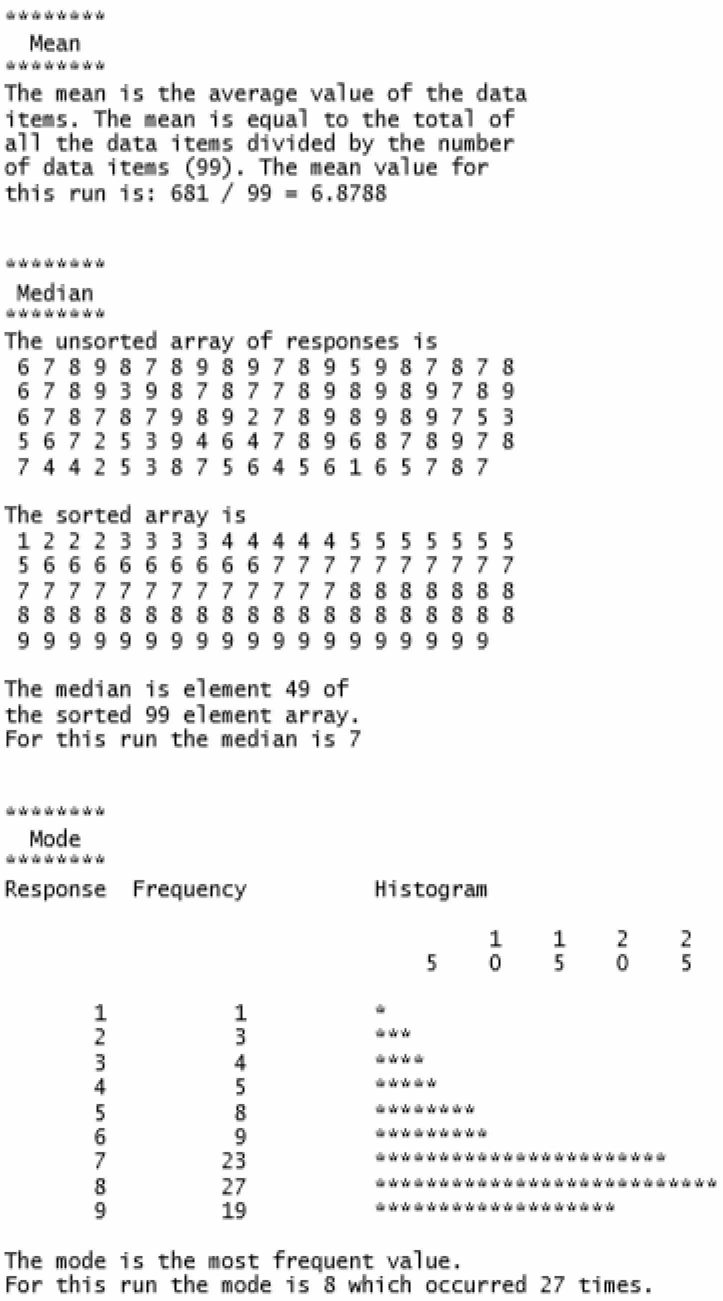
Your answer need not be in the lowest terms.

2. Modify Q1 for homework and show the answer in lowest forms (not in decimals. Only answers in fractions are allowed). For example, if answer is 6 / 15, you will show 2 / 5.
3. Yasir has some raw data with total number of items equal to 99. He needs to find how data is distributed. For this he needs to find mode, mean and median of the given data and then plot its histogram (A graph which has item on x-axes and its frequency on y-axis. See Wikipedia for more details) for visual aid. Write modular code for this problem. Modular means think about the possible functions/modules which are required to be created for this problem. Each function should be separable from the second one or runs independently of the other. It will be considered as a black box which takes some inputs and then outputs. Helper functions can also be used.
   You must also note that the data of 99 integer items must be produced randomly. You will not ask it from user instead generate it in your code. For example, for the given data:



```
{6, 7, 8, 9, 8, 7, 8, 9, 8, 9,
 7, 8, 9, 5, 9, 8, 7, 8, 7, 8,
 6, 7, 8, 9, 3, 9, 8, 7, 8, 7,
 7, 8, 9, 8, 9, 8, 9, 7, 8, 9,
 6, 7, 8, 7, 8, 7, 9, 8, 9, 2,
 7, 8, 9, 8, 9, 8, 9, 7, 5, 3,
 5, 6, 7, 2, 5, 3, 9, 4, 6, 4,
 7, 8, 9, 6, 8, 7, 8, 9, 7, 8,
 7, 4, 4, 2, 5, 3, 8, 7, 5, 6,
 4, 5, 6, 1, 6, 5, 7, 8, 7};
```

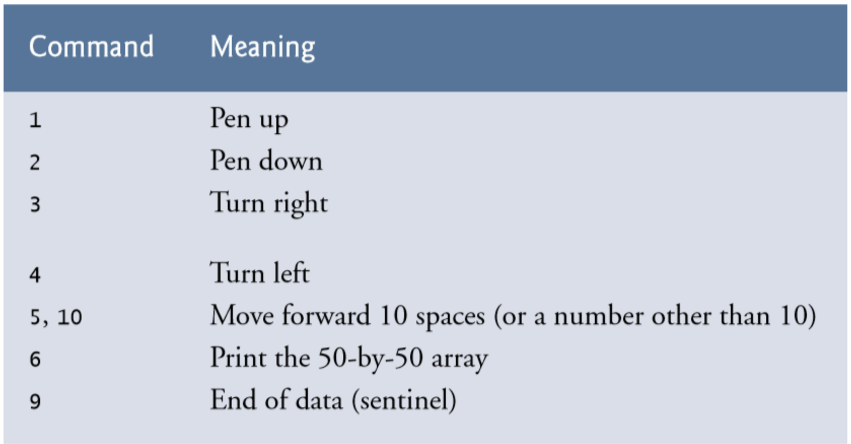
   Output will be



```
*********
  Mean
*********
The mean is the average value of the data
items. The mean is equal to the total of
all the data items divided by the number
of data items (99). The mean value for
this run is: 681 / 99 = 6.8788


*********
 Median
*********
The unsorted array of responses is
 6 7 8 9 8 7 8 9 8 9 7 8 9 5 9 8 7 8 7 8
 6 7 8 9 3 9 8 7 8 7 7 8 9 8 9 8 9 7 8 9
 6 7 8 7 8 7 9 8 9 2 7 8 9 8 9 8 9 7 5 3
 5 6 7 2 5 3 9 4 6 4 7 8 9 6 8 7 8 9 7 8
 7 4 4 2 5 3 8 7 5 6 4 5 6 1 6 5 7 8 7

The sorted array is
 1 2 2 2 3 3 3 3 4 4 4 4 4 5 5 5 5 5 5 5
 5 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7
 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8
 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

The median is element 49 of
the sorted 99 element array.
For this run the median is 7


*********
  Mode
*********
Response  Frequency          Histogram

                                      1    1    2    2
                                 5    0    5    0    5

       1          1          *
       2          3          ***
       3          4          ****
       4          5          *****
       5          8          ********
       6          9          *********
       7         23          ***********************
       8         27          ***************************
       9         19          *******************

The mode is the most frequent value.
For this run the mode is 8 which occurred 27 times.
```

4. The Logo language made the concept of turtle graphics famous. Imagine a mechanical turtle that walks around the room under the control of a C++ program. The turtle holds a pen in one of two positions, up or down. While the pen is down, the turtle traces out shapes as it moves; while the pen is up, the turtle moves about freely without writing anything. In this problem you'll simulate the operation of the turtle and create a computerized sketchpad as well. Use a 50-by-50 array floor which is initialized to zeros. Read commands from an array that contains them. Always keep track of the current turtle position and whether the pen is currently up or down. Assume that the turtle always starts at position 0, 0 of the floor with its pen up. The set of turtle commands your program must process are shown in Figure. Suppose that the turtle is somewhere

near the center of the floor. The following "program" would draw and print a 12-by-12 square. As the turtle moves with the pen down, set the appropriate elements of array floor to 1s. When the 6 command (print) is given, wherever there's a 1 in the array, display an asterisk, or some other character you choose. Wherever there's a zero, display a blank. Write a program to implement the turtle graphics capabilities discussed here. Write several turtle graphics programs to draw interesting shapes. Add other commands to increase the power of your turtle graphics language.



Command	Meaning
1	Pen up
2	Pen down
3	Turn right
4	Turn left
5, 10	Move forward 10 spaces (or a number other than 10)
6	Print the 50-by-50 array
9	End of data (sentinel)

iv) Submission instructions:

- You should create separate file for each question with proper names i.e. q1, q2 and qn etc. Zip them and submit them on Eduko according to the date.