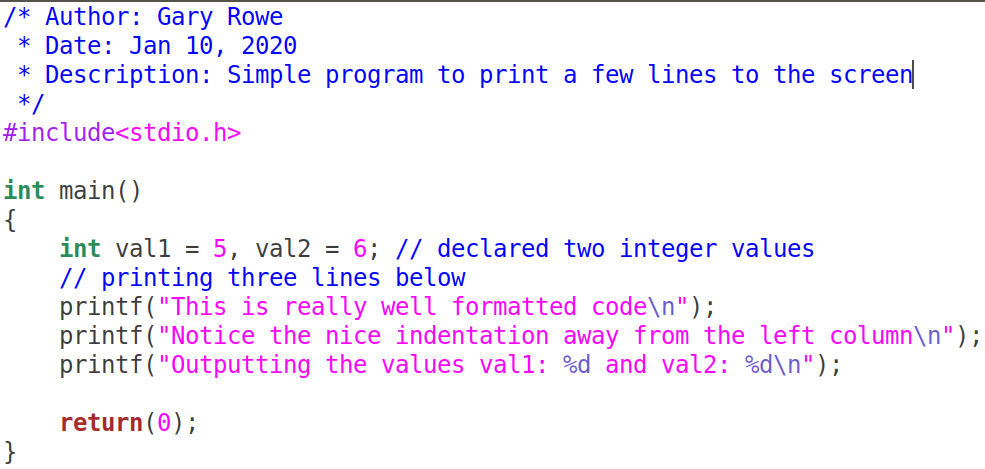
**Student Name:** **Weight:**

**Student ID:** **Marks:** \_\_\_\_**/80**\_

**CREATE AN EMPTY DOCUMENT TO SUBMIT YOUR SOLUTIONS. DO NOT USE THIS DOCUMENT TO SUBMIT YOUR ANSWERS. YOU WILL LOSE 10% FOR DOING SO!!!**

Assignment 1:

Your C files should be properly ***formatted*** with ***indentations*** that enhance code ***readability***. Example of properly formatted code:

**Important:**

* On your Linux guest machine use the folder created in class to store the files.
* Inside the ITSC202 folder create a subfolder called **A2**.

A1 is the folder you will use for all the C files for **Assignment 2**.

**Problems 1 and 2 HINT**:

Imagine you are drawing on grid paper.

* You start at row 0 and draw all the columns on that row
* Then you go to row 1 and you draw all columns on that row etc
* Next think about what condition must be met to draw a star at the column position.
* Are there any rows that look the same? What logic can you use to simplify those rows?

Problem 1 (10):

Write a program that prints the pattern shown below:

The criteria are as follows:

* In the folder Assignment 2, create a file called m2p1.c.
* Use loops (for/while) and if statements
* Use printf("**\***"); printf("**\n**"); and printf(""); statements to complete this question.
* ONLY 2 loops are allowed. They should be nested loops.

**You will submit the C source code and a sample output of the code. Use screenshots, don’t copy paste as the output will not look as it should.**

[/home/myname/Assignment2/]$./**m2p1**



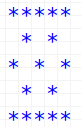
Problem 2 (15):

Write a program that prints the pattern shown below:

The criteria are as follows:

* In the folder Assignment 2, create a file called m2p2.c.
* For a challenge try to complete it with a single **printf("\*");**
* ONLY 2 loops are allowed. They should be nested loops.

**You will submit the C source code and a sample output. Use screenshots, don’t copy paste as the output will not look as it should.**

[/home/student/Assignment1/]$./**m1p2**

Problem 3 (15):

You will write a program that takes as input 10 integer numbers. The program will do the following with the numbers entered:

1. Print the average of the numbers entered
2. Print the sum of the numbers entered
   1. If the sum is even, print a message saying so.
   2. If the sum is odd, print a message saying so.
3. The average MUST be shown with no more than 2 decimal places.
   1. Google: How to display only 2 decimal places in C

**Special Notes:**

* The code should **NOT** use arrays.
* You will use loops and if statements to complete this assignment.
* If possible complete the program with no more than 4 variables.

**You will submit the C source code and a sample output of the code that looks similar to the below table.**

|  |  |
| --- | --- |
| [/home/myname/Assignment2/]$./**m2p3**  Please enter number 1: 1  Please enter number 2: 2  Please enter number 3: 3  Please enter number 4: 4  Please enter number 5: 5  Please enter number 6: 6  Please enter number 7: 7  Please enter number 8: 8  Please enter number 9: 9  Sum is: 45  The number is odd  The average is: 5.00 | [/home/myname/Assignment2/]$./**m2p3**  Please enter number 1: 1  Please enter number 2: 2  Please enter number 3: 1  Please enter number 4: 3  Please enter number 5: 43  Please enter number 6: 45  Please enter number 7: 12  Please enter number 8: 65  Please enter number 9: 78  Sum is: 250  The sum is even  The average is: 27.78 |

Problem 4 (15):

Copy the following C code snippet. Complete it according to the criteria below. Once completed you will execute the code and answers and fill in the table:

Criteria to complete:

* You will print the address of each variable using the & operator. The address format: (0x121YYY). **Y** - represents some hexadecimal number

**HINT**: What is %p used for?

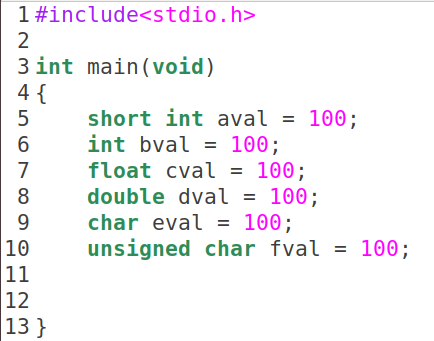
* You will print the sizeof of each variable using the sizeof operator.
* Print the content of each variable based on the **datatype**
* **Example output of your code:**

**Address of aval: 0x121YYY, Value of aval: 100, sizeof aval: M bytes**

**Analysis:**

1. Appropriately comment your code.
   1. **Submit the commented code and the answers to the questions below**
2. Answer the following questions:
   1. Complete the following table based on the output of your code:

|  |  |  |
| --- | --- | --- |
| Variable name | & of variable | Content of variable |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |



Problem 5 (25):

You will create code that follows the alogorithm below **(DO NOT USE ANOTHER ALOGORITHM)**:

If the original number is aval, where aval is assumed to be less that 255:

|  |  |  |  |
| --- | --- | --- | --- |
| Step no. | Operation | Result | Remainder |
| 1 | Aval / 2 | Res1 | Rem1 where ( 0 rem1 1 ) |
| 2 | Result1 / 2 | Res2 | Rem2 |
| 3 | Result2 / 2 | Res3 | Rem3 |
| 4 | Result3 / 2 | Res4 | Rem4 |
| 5 | Result4 / 2 | Res5 | Rem5 |
| 6 | Result5 / 2 | Res6 | Rem6 |
| 7 | Result6 / 2 | Res7 | Rem7 |
| 8 | Result7 / 2 | Res8 | Rem8 |

Result would then be written as:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rem8 | Rem7 | Rem6 | Rem5 | Rem4 | Rem3 | Rem2 | Rem1 |

Where RemX (X is 1-8) is equal to either 0 or 1.

Example, take aval to be 130:

|  |  |  |  |
| --- | --- | --- | --- |
| Step no. | Operation | Result | Remainder |
| 1 | 131 / 2 | 65 | 1 |
| 2 | 65 / 2 | 32 | 1 |
| 3 | 32 / 2 | 16 | 0 |
| 4 | 16 / 2 | 8 | 0 |
| 5 | 8 /2 | 4 | 0 |
| 6 | 4 / 2 | 2 | 0 |
| 7 | 2 / 2 | 1 | 0 |
| 8 | 1 / 2 | 0 | 1 |

The resulting binary number would be:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rem8 | Rem7 | Rem6 | Rem5 | Rem4 | Rem3 | Rem2 | Rem1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Analyze the algorithm on the previous page and write a C program to generate the resulting binary number:

Criteria:

1. **Do not use arrays to complete this**
2. Print the binary result to the screen, with the most significant bit on the **RIGHT**.
3. The program should run in a loop until the user enters **a number larger than 65535**
4. Your output should match the output below:

[/home/myname/Assignment2/]$./**m2p5**

Please enter a number between 0 and 65535: 45000

0001 0011 0000 0101

[/home/myname/Assignment2/]$./**m2p5**

Please enter a number between 0 and 65535: 32456

0001 0011 0111 1110

[/home/myname/Assignment2/]$./**m2p5**

Please enter a number between 0 and 65535: 65432

0001 1001 1111 1111

IMPORTANT: If you are having difficulty with printing the exact output but you are able to print the output based on the exact algorithm, you will get part marks.

Do not spin your wheels indefinitely, ask for help if you need it.

You **MUST** use this exact algorithm.