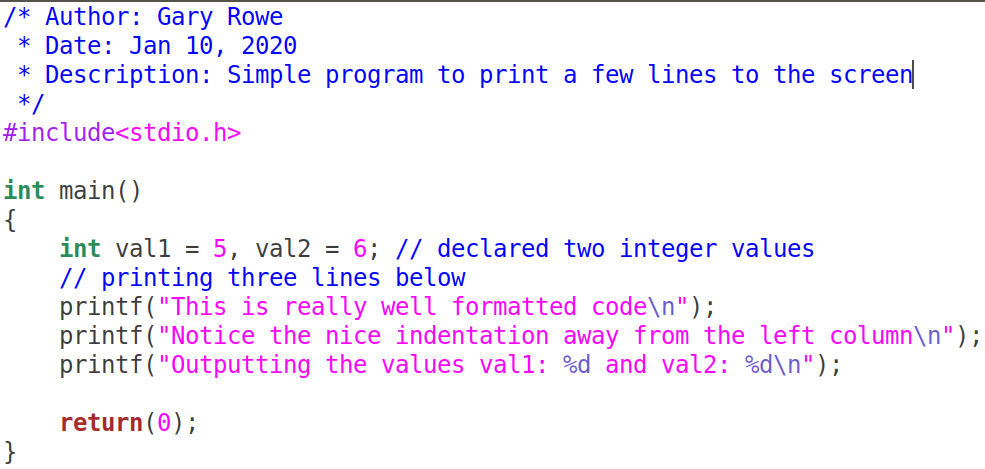
**Student Name:** **Weight: 50**

**Student ID:** **Marks: \_\_\_\_\_\_\_\_\_\_**

**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 6:

Your C files MUST be properly ***formatted*** with ***indentations*** that enhances your code ***readability***. Example of properly formatted code:

Problem 1 (25):

**Write a program that will take command-line arguments from the user and perform the operations as shown in the examples:**

**./cmdline “hello ” + “World ” + “Good Times”**

***hello World Good Times***

**./cmdline1 “Hello” – “el”**

***Hlo***

**./cmdline1 “Hello” \* 3**

***HelloHelloHello***

**./cmdline1 “Hello” + “you” - “loy”**

***Helou***

**./cmdline1 “Hello” - “boot” + “ Eloy”**

***Hello Eloy***

**Requirements**:

1. You must have an array that contains the final string to be printed. This string gets printed after all the other operations have been completed.
   1. **There must be an array that contains the final variable.**
2. You can use functions that check if the string to be removed is a substring; and returns the offset of the beginning of that string.
3. You will need to have a method to append strings to another string. All strings are added to the end.
4. Removing a substring will cause the other parts of the string to shift forwards to find the space.
   1. **Only the first occurrence of the substring to be removed is considered**.
5. **You must also modularize your code**. You must have functions to perform the operations required.
6. Strings operations can include the following: (addition, subtraction, and multiplication).
7. **Multiplication must be performed first**.
8. If a string does not contain the substring to be removed the resultant string is the first string.

**Requirements for submission:**

1. Your completed code
2. Screen capture of the code running. Show at least two scenarios of your code working.

All functions are available for use and there are no restrictions on the functions you can use. Example strncat, strcat, strstr etc.

Problem 2 (25): (Break down into manageable chunks [use functions])

**Write a program that will take the the following command line arguments:**

**./cmdline2 -b 100 -e 200 -s 4 -f textfile.txt -m w**

**Characteristics of the program:**

* Name of executable **cmdline2**
* **-b** represents the beginning number of a range
* **-e** represents the end number of a range
* **-s** represents the step size of the range
* **-f** represent the file to read from or write to
* **-m** represents the mode of operation (**w**)rite or (**r**)ead

1. The program will print to the screen, the range of numbers based on the above options. It will write that same output to the file textfile.txt.

**./cmdline2 -b 100 -e 200 -s 4 -f textfile.txt -m w**

***100 104 108 112 116 120 124 128 … 188 192 196 200***

1. The program will read the range of numbers placed into the file based on the command line parameters.
   1. It will also print those values to the screen.
   2. If the mode is r for read, ignore all other options except -f for the file name.

**./cmdline2 -f textfile.txt -m r**

***100 104 108 112 116 120 124 128 … 188 192 196 200***

1. If any options is missing the program does not run

**eg ./cmdline2 -b 100 -s 4**

***Range end value is missing please try again***

1. If the step size does not produce at least 1 iteration of the loop, the program doesn’t run

**eg** **./cmdline2 -b 100 -e 200 -s 200**

***Program can’t run step size doesn’t allow program to iterate once.***

**Requirements for submission:**

1. Your completed code
2. Screen capture of the code running. Show at least two scenarios of your code working.

You can use any library functions you require. You must also modularize your code; don’t put all of your logic in main.