Honors Computer Science C++
Mr. Clausen
Program 12A & 12B

Special Note: Every program from Chapter 4 to the end of the year needs to have functions!

PROBLEM 12A: Arithmetic Series (20 points)

Write a program that **uses recursion and one recursive call only (there cannot be any other loops of any type anywhere in your program, otherwise your program will receive a grade of half credit)** to generate an Arithmetic Series of the numbers from 1 to 100 with running sums of the numbers. (On line one the sum of just 1. On line two the sum of 1+2, line 3 sum of 1+2+3, line 4 the sum of 1+2+3+4, line 5 the sum of 1+2+3+4+5 etc.) Use a recursive function named Sum_Up to do the work. Print out the results in a nice table format as illustrated below.

```
TERM #          SUM
= = = = = = = = = = = =
1                 1
2                 3
3                 6
4                10
etc..
```

Save the program as LastNameFirstNameP12A.cpp in your "S:" directory. To see a model for this program look at the source code for the program "recursn2.cpp" or "recursn3.cpp" in the network directory titled: HonorsCompSciCFiles. Look in the folder, Recursion.
As you type all your programs this year, be sure not to type past the 80-column line in Borland C++ 5.02 for Windows. If you have any statements longer than 80 columns, press the return key to "wrap" the statement around to the next line.

1. Type comments at the beginning of the program to display your name and other information just like those used for program 2A. **Make sure to change the program name and program description in these comments, so that the program number, name, and description say what is listed above.**

2. Include <iostream.h> so we can use the **cout** and **cin** commands. Also include <conio.h> so you can use **getch**() to leave your output displayed on the screen until the user presses any key to continue. Include <iomanip.h> so we can use the **setw** command.

3. There are no constants to declare.

4. Don't forget to include comments for your functions, right before each of your function declarations.

5. Declare the function Display_My_Info ( ) in the function declaration section of the program. This should be a void function that doesn't take any parameters and doesn't return any values. In addition, declare a function for Display_Heading(), that is a void function with no parameters that simply prints the heading for your output, and Sum_Up(), that should be a void function also, but requires parameters for a counter and a sum. Sum_Up( ) should be a "tail recursive function" (see the lecture notes).

6. Use a comment line with **equal signs** to separate all of the above from the int main function. For example:
   //==========

7. Inside the **int** main() function, declare all of your variables and then call the functions. **Remember that recursive functions call themselves after they have been called originally from int main. Don't forget that value-returning functions are "called" in assignment statements, while void functions can be called just by writing the function name including any of the "actual" parameters.** Then tell the user to press any key to continue, have your getch() command, and your return 0 command.

8. After the **int** main() function, have another comment line of **subtraction signs** to separate the above from your function implementation lines below. For example:
   //----------

9. Implement all of the functions, separating each one from the other ones using comment lines of **subtraction signs**.

10. When you implement the function: Display_Heading(), create a "heading" for these numbers and have the numbers listed in two columns on the screen. To format the numbers nicely use the setw command to set a width for each of the columns. You also want to right justify these numbers. This will require that you add #include <iomanip.h> to your preprocessor directives and use the following commands: cout << setiosflags (ios::right) ; and cout<< setw(choose your number here);

11. After the last function implementation of every program, end your program with two comment lines of equal signs. This signifies the end of your source code. For Example:
    //==========
    //==========

When you are finished with your program, have tested it thoroughly to make sure that your calculations are correct, and are sure that you don't need to make any changes, then save your program in the "W" network mapping, and the Program 12A folder.

PROBLEM 12B: Sorts and Searches Part 2 (24 points)

Finish program 8A by adding the rest of the searches and sorts. This will mean adding the Binary Search, the Quick Sort, and the Merge Sort.