Lab 4 Recursion

Goals-
Create recursive functions
Demonstrate your functions in a program

1. Write a function that recursively reverses a string. The function will have a single parameter of type string. It will print the reversed characters to the screen followed by a newline character. Before you start writing code answer these questions. What is the base case? Will you print out the characters before or after the recursive function call? HINT: You can try it both ways and see if it makes a difference.

2. Write a function that recursively calculates the sum of an array. The function should have 2 parameters, the array and an integer showing the number of elements in the array. The function will use a recursive call to sum the value of all elements. What is the base case? What is the structure of the recursive call?

3. A triangular number is the sum of all items that form a triangle with N rows. It may help to think of the number of bowling pins if you changed the number of rows. For example in a standard game there are 4 rows and 10 pins. If you add a fifth row you add 5 pins for a total of 15. Always remember mathisfun.com!

Write a recursive function to calculate the triangular number for N. (Or a function that counts the number of pins necessary for N rows.) What is the base case? How many items are in row 1? How many are in row 2? How do you sum the number of items in each row and add it to the total value?

You must demonstrate your functions in a program. For #1 you must prompt the user for a string and your program will print the reversed string. For # 2 you must prompt for a series of numbers to enter in the array. As you read the numbers increment a count so you know how many are in the array. How does the user indicate they are finished entering numbers? Then your program will print the sum. For #3 prompt the user to enter an integer and your program will return the total number of items in the triangle with that number of rows.

You will have a single program to demonstrate all three recursive functions. Provide a menu for the user to select which function to run, and then return to the menu. You will need a fourth item in the menu to allow them to exit the program.

Modular Grading

We are using modular grading. Each lab will be divided into specific modules. Each module will be graded pass/fail. It either works properly or it does not. 10% of every lab or assignment grade is style/comments or other elements of self-documenting code and clarity. Remember the labs are worth 10 points total.

Programming style- 1 point

You implement the function to reverse a string and demonstrate in a program- 2 points

You implement the function to calculate the sum of the array and demonstrate in a program- 3 points

You implement the function to calculate the triangular number and demonstrate in a program- 3 points

You implement a program with a menu to call each function or to exit- 1 point