

(60-140) Lab Exercises #9

(Due at the end of the lab period or beginning of the next)

Nov. 22-25, 2016.

Objective: Working with iterative and recursive functions

Triangular Numbers: If you arrange 15 dots in the shape of a triangle, you end up with an arrangement that might look something like this:

```
  .  
 . .  
. . .  
. . . .  
. . . . .
```

The first row of the triangle contains one dot; the second row contains two dots, and so on until the fifth row contains five dots. Whenever a row has more than one dot, there is a space between dots. As shown in the example that $n = 5$ has 15 ($=1+2+3+4+5$) dots, the number of dots it takes to form an n -row triangle in general is the sum of the integers from 1 through n . This sum is known as a **triangular number**.

Develop a program to first take a positive number n (greater than 0) from the user, and then calculate and output the value of the triangular number for n . Your design and implementation of the program must have one function to iteratively calculate the number with loop structure, and another function to obtain the same result but by calling the function itself recursively.

Your program should have at least the following 3 functions (procedures):

- 1) Use `readNum()` to take a valid number (greater than 0) from the user. This function has no input and returns a valid integer number n entered by the user.
- 2) Use `iterativeTriangularNumber(...)` to calculate triangular number for n iteratively. This function has one integer variable as an input, named `number`, and returns the value of the triangular number.
- 3) Use `recursiveTriangularNumber(...)` to calculate the triangular number for n recursively. This function has one integer variable as an input, named `number`, and returns the value of the triangular number.

Hint: As a base case, if `number` is equal to 1 then the triangular number of 1 is 1.

A Sample interaction is as follow:

```
Enter a valid number (>0): -1  
Invalid number.  
Enter a valid number (>0): 5  
Iterative: Triangular number of 5 is 15.  
Recursive: Triangular number of 5 is 15.
```

Part A: RAPTOR Exercise

- a) Start RAPTOR and create the flowchart of Triangular Numbers.
- b) Save the flowchart to a file named "triangularNumbers.rap" in the working directory on the PC you are using.
- c) Demonstrate the triangularNumbers.rap file to GA/TAs and run with different input values.

Hint: The mode of your Raptor should be "Intermediate" to be able to add a procedure.

Part B: C Programming Exercise

- a) Implement the algorithm as represented by "triangularNumbers.rap", and write an equivalent C program that accomplishes what the flowchart does.
- b) Save your program to a file named "triangularNumbers.c" in the working directory on the PC you are using.
- c) Demonstrate the triangularNumbers.c file to GA/TAs and run with different input values.

EVALUATION:

You need to show your GA/TA the complete programs at the end of this lab, or at the beginning of your next lab. The marks you will receive for this lab are made of two parts: Lab work marks 10 and attendance marks 5. **Total 15 marks.**

Lab Work Mark: Your C code will be evaluated based on your solutions for the problems based on the following scheme:

1. Does the code run and meet specifications?
 - Is input adequate and input data type properly validated?
 - Is processing adequate?
 - Is output correct and adequate?
 - Is the code compliant? Is the code run properly?
2. Is the code properly commented?
 - Is the program title, programmer's first and last name, and the date posted at the top in a multi-line comment?
 - Is each significant step of the program properly commented?
 - Are comments added to clarify details?
 - Are comments clear, accurate, neatly formatted, and have no misspellings?
3. Is the code properly formatted?
 - Are blocks of code indented according to their parent-child relationship?
 - Do curly braces line up vertically?
 - Is there an empty line between significant steps (blocks) of the program?
 - Is the width of the code contained within a reasonable limit so that minimal horizontal scrolling is required (with 800 x 600 monitor resolution), and there is minimal line-wrapping when printed?
 - Is camel-case notation used for variable, e.g. employeeLastName?

IMPORTANT: DO YOUR OWN CODE. ANY CODE SUSPECTED TO BE SIMILAR TO ANOTHER SUBMISSION WILL CAUSE BOTH SUBMISSIONS TO RECEIVE A ZERO MARK ON ALL LABS AND BE REPORTED FOR PLAGIARISM.