CS 211 – Data Structures
Lab for Week #03

For this lab exercise, you will work either in 2-person teams or individually. You are HIGHLY ENCOURAGED to work in 2-person teams! The point here is to have teams discuss the lab and offer each other support in making sure the IDE of your choice is working.

Lab Exercise

This lab is to review the principles of recursion, and to practice writing and using inductively-structured recursive functions. Download the supplied **main.cpp** and **week03Lab** files, and interactively test your functions. Write all function code in the single CPP file named **week03Lab.cpp**. Make sure the function headers match the ones in the **week03Lab.h** file!

- Define and test a function **int sumDownBy2(unsigned int n)** that **uses recursion** (not a loop!) and returns the sum of positive integers **n**, (**n-2**), (**n-4**), etc., down to zero or one, whichever is appropriate. Note that the recurrence relation that does this calculation is:
  sumDownBy2(n) = n + sumDownBy2(n – 2)
- Define and test a function **int recursiveMult(unsigned int j, unsigned int k)** that **uses recursion** (not a loop!) and returns the product of **j** and **k** **without using the multiplication operation** but instead makes use of only addition and recursive calls.
  For this one, here's a hint:
  * Write a recurrence relation that relates the value of **j*k** to the value of **j*(k-1)**
- Define and test a function **bool isPalindrome(string inputString)** that expects a string of any length and returns true if the string is a palindrome (that is, a string that is the same when its characters are written in reverse order). Assume lower-case letters only. Your code **MUST implement recursion**!
- Define and test a function **double geometricSum(unsigned int n)** that **uses recursion** to calculate and returns the value of $1 + ½ + ¼ + ... + (½)^n$. Note that this sum can be expressed as
  $\sum_{i=0}^{n} (\frac{1}{2})^i$
  Do NOT use an exponent operation – instead make use of only addition and division and recursive calls. (HINT: If what you have is the value of the sum from i=0 to n-1, you can calculate the sum from 0 to n using a single division and a single addition. This may require a bit of creative thinking!)
  This one is tricky! What are the same exact two identical steps that get you from, say, 1 to $1 + ½$ that also gets you from $1 + ½$ to $1 + ½ + ¼$? Use that to help write your recurrence relation:
  geometricSum(n) = < some expression that contains geometricSum(n-1) >
- At the top of your team's **week03Lab.cpp** file, include these lines:
  `// CS 211 Fall 2024 – Week 03 Lab`
  `// <Student_Name1> and <Student_Name2>`

ONCE FINISHED, ALL STUDENTS SHOULD SUBMIT VIA CANVAS.
You should send your CPP file to all team members for submission.
Submit ONLY the week03Lab.cpp file via Canvas.
Do NOT submit the main.cpp or week03Lab.h files – the instructor already has those!