**Module 4A and B - Assignment**

For the problem statement below, you must write the Pseudocode for each part of the solution. The Pseudocode must be complete and following the standards listed at [http://ccse.kennesaw.edu/fye/Pseudocode.php](https://kennesawedu-my.sharepoint.com/personal/dtatum7_kennesaw_edu/Documents/9.%20Capstone%20project%20final%20acceptance%20and%20assessment%20Hackthon%20Registration%20-%20Dawn%20Tatum.docx?web=1)http://ccse.kennesaw.edu/fye/Pseudocode.php

After completing the pseudocode, please use it to create the source code in your lab’s specific language (either C# or Java). Make sure to turn in both the pseudocode and source code to Gradescope and in their appropriate drop boxes. The pseudocode will be turned into assignment 4A and the source code to 4B.

**Problem Statement**:

Write a program that uses a recursive method to convert a number in decimal to a given base b, where b is between 2 and 36. Your program should prompt the user to enter the number in decimal and the desired base.

Hint: The algorithm to convert a positive decimal number into an equivalent number in octal (or hexadecimal) starts by dividing the decimal number by 8 (for octal) and by 16 (for hexadecimal). Suppose that ab represents a to the base b. Then 7510 represents 75 base 10 (decimal) and 7516 represents 75 base 16 (hexadecimal). Check out these equivalents:

75310 = 13618

75310 = 2F116

You can extend the idea of converting from a decimal to base 2, 8 or 16 to any base. Suppose you wish to convert a decimal number n into an equivalent number in base b, where b is between 2 and 36. You would start by dividing the number n by b. Note: the digits in say base 20 are 0,1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H, I and J.

You need to write your program and test it using the test cases below.

Test Data Inputs:

9098, base 20

692, base 2

753, base 16

**Break the problem down and write the pseudocode:**

Break the problem statement into parts to fully understand what is expected as input, output and processes in your code. Write down all the variables you will need from the problem statement. Once you have broken down the problem, start writing your pseudocode.

Writing the pseudocode may take a while and several iterations to fully get the solution complete. Walk through the problem and your pseudocode as if the program is running.

Now make sure that your code is the best it can be. Check your formatting, comments, and make sure your output is correct and communicates what is happening in your code.

**What to Turn In**

Follow submission guidelines on this page: <http://ccse.kennesaw.edu/fye/Submission%20Guidelines.php>