CSE 1321L: Programming and Problem Solving I Lab

Assignment 1 – 100 points

Pseudocode Algorithms

Note: Solutions for this assignment must be written in Pseudocode using the Pseudocode Template (one template per problem) and following the Problem Solving Guide from our website (under Pseudocode). See CCSE FYE Website for more information about Pseudocode.

What students will learn:

1) Problem solving
2) Basic program structure
3) Input and output with the user
4) Basic calculations

Academic Policies: don't cheat. Come see us instead – we're here to help.

Problem 1: Red, Green or Blue? One of the truly fascinating things about the world is that it's *infinite*. What do we mean by this? As an example, we could ask you how many shades of yellow there are, and you'd soon realize there are an infinite number of them. Because computers don't have an infinite amount of storage space, they have to approximate the natural world. For colors, they most often use an *additive color model* comprised of (R)ed, (G)reen and (B)lue. They store these values either as a floating point number 0.0 – 1.0 (fully off to fully on), or an integer between 0 to 255. Your task is to:

a) Ask the user to individually enter in three values between 0-255
b) Read those three values in
c) Convert each number to a new range of 0.0 – 1.0
d) Display the results

The algorithm output is as shown below, with user input in bold. Can you guess the color the user entered?

Sample Output:

```
Enter a red value (0-255): 101
Enter a green value (0-255): 67
Enter a blue value (0-255): 33
New color is red=0.396, green=0.263, blue=0.129
```

Program 2: You're McAwesome™! Imagine a certain McHamberger® place asks for your McHelp©. They are piloting a new meal called the "McGrease®™ Meal" that contains a hamburger, French fries and a drink. They want you to calculate how much profit they are going to make during this campaign. Your task is to write a pseudocode program that asks the user for 1) the cost of the meal, 2) the amount the meal will sell for, 3) the number they predict they will sell and 4) the percent you have to give back to corporate (i.e. the restaurant chain). You should then calculate how much profit McHamberger® Inc. will make after paying tax.

Sample Output:

```
Enter the cost to make the meal: 0.50
Enter how much you will sell the meal for: 7.25
Enter how many you think you will sell: 50000
Enter the percent to return to corporate: 0.06
This campaign will make $317250
```

Sample Output #2:

```
Enter the cost to make the meal: 3.00
Enter how much you will sell the meal for: 4.25
Enter how many you think you will sell: 100000
Enter the percent to return to corporate: 0.07
This campaign will make $116250
```

Program 3: Assume that the McHamberger™ place from above found out that the McGrease® meal caused excessive weight gain. In order to help their reputation, they are now offering a program that determines how much weight a person can gain or lose in a given amount of time. Assume that the average person can eat 2,500 calories per day without gaining or losing weight. Your task is to write a program that asks the user for the number of calories they plan on eating per day as well as the number of days they will do that. It then displays the total weight the user will gain or lose. Note: there are 3,600 calories in one pound. Of course, legally, consult your physician about any dietary changes you may be considering ☺

Sample Output:

```
Enter the number of calories you will consume per day: 2600
Enter the number of days you will eat this much: 36
Weight change is 1.0 lbs
```

Sample Output #2:

```
Enter the number of calories you will consume per day: 4000
Enter the number of days you will eat this much: 365
```

```
Weight change is 152.0833 lbs
```

Sample Output #3:

```
Enter the number of calories you will consume per day: 1000
Enter the number of days you will eat this much: 200
Weight change is -83.3333.0 lbs
```

Program 4: But I haven't taken Calculus yet! OK – we admit, this one might look nasty, but read on because it's not that bad. One of the things that computing folks do A LOT (especially CS students) is analyze how long a program takes to run. In the first step, something might take 1 unit of CPU time. The second step might take 2, the third 3 and so on. Mathematically, that would be expressed by the "summation" operator as this:

$$\sum_{i=1}^{n} i$$

This is may look complex, but all it's saying is that i goes from 1 to n and you sum all those numbers together. So, if (n = 10),

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$$

Fortunately, this reduces to a simple formula. When summing numbers from 1-n, the sum is:

(n+1)*n/2

Why? It's all in how you "pair" the numbers together. In the example above, we could pair (1+10)=11, (2+9)=11, (3+8)=11 and so on. That is, we have n/2 "pairs" of (n+1).

Your task is, without using a loop (which we haven't covered yet), to write a program that asks the user for a number, then prints out the sum of 1 to that number. If you use a loop, you will receive no credit.

Sample Output:

```
Enter a number: 3
The sum of the numbers between 1 and 3 is 6
```

Sample Output #2:

```
Enter a number: 10
The sum of the numbers between 1 and 10 is 55
```

```
Sample Output #3:

Enter a number: 15
The sum of the numbers between 1 and 15 is 120
```

Program 5: I want an A! Write a pseudocode algorithm that asks the user for their grades in CSE 1321, along with their attendance, then calculates their final grade. Note, perfect attendance rounds up your final grade by 1.5 points – otherwise, it's a fraction out of 30. The tests and quiz average are worth 20% each.

```
Sample Output:

Enter your grade for Test 1:  95
Enter your grade for Test 2:  85
Enter your grade for Test 3:  63
Enter your grade for Test 4:  70
Enter your average quiz grade: 81
Enter the number of times you attended class (max 30): 29
Your average before attendance is 78.8.  You receive an
additional 1.45 points for attendance.  Final grade is 80.25.
```

Submission:

1. Review the assignment submission requirements and grading guidelines.
2. Upload the pseudocode files (Word doc or PDF) to the assignment submission folder in Gradescope.
3. The files must be uploaded to Gradescope by the due date.

[and yes, we intentionally misspelled "Hamberger"]