ECE 71 – Introduction to Computational Programming

Professor Kriehn – Fall 2017

Code Due By: Midnight on Tuesday, September 19, 2017
Writeup Due By: Class on Wed/Thu, September 20/21, 2017

HOMEWORK #07 – Currency Bill Verification System

You work at a bank and needs to automate the bills that are given to a customer when they cash a check. When the check is cashed, the system should automatically break down the dollar amount into the smallest number of $20, $10, $5, and $1 bills.

Specifications:
Write a program that prompts the user to enter a U.S dollar amount and then shows how to pay the customer using the smallest number of bills. After the program runs, prompt the user if he/she needs to enter another dollar amount, and repeat as necessary.

After prompting the user to enter the dollar amount, scan it in and calculate the number of twenties, tens, fives, and ones that are necessary to pay to the customer. You may assume that the dollar amount is always an integer value.

Since the program should allow the user to input multiple dollar amounts, use a do-while loop to help solve the problem.

As an example, if you execute the program with the following underlined inputs, the output will be:

```
~> main.o
Enter a dollar amount: 93

$20 bills: 4
$10 bills: 1
$ 5 bills: 0
$ 1 bills: 3

Convert another dollar amount? (Y/N): Y
Enter a dollar amount: 42

$20 bills: 2
$10 bills: 0
$ 5 bills: 0
$ 1 bills: 2

Convert another dollar amount? (Y/N): N
~>
```

Develop your I/O diagram and pseudocode, debug your code, and submit to the Grader Program.

ECE 71 – Introduction to Computational Programming

Professor Kriehn – Fall 2017

HOMEWORK #08 – Weather Balloons

Weather balloons are used to gather temperature and pressure data at various altitudes in the atmosphere. The balloon rises because the density of the helium inside the balloon is less than the density of the surrounding air outside the balloon. As the balloon rises, the surrounding air becomes less dense, and thus the balloon's ascent slows until it reaches a point of equilibrium. During the day, sunlight warms the helium trapped inside the balloon, which causes the helium to expand and become less dense; thus, the balloon will rise higher. During the night, however, the helium in the balloon cools and becomes more dense; thus, the balloon will descend to a lower altitude. The next day, the sun heats the helium again and the balloon rises. Over time, this process generates a set of altitude measurements that can be approximated with a polynomial equation. Assume that the following polynomial representation represents the altitude or height in meters during the first 48 hours following the launch of a weather balloon:

$$h(t) = -0.12t^4 + 12t^3 – 380t^2 + 4100t +220,$$

where the units of t are in hours. The corresponding polynomial model for the velocity in meters/hr of the weather balloon is:

$$v(t) = -0.48t^3 + 36t^2 -760t + 4100.$$

Write a program that will print a table of the time, altitude, and the velocity for this weather balloon using units of meters and meters per second. Let the user enter the start time, the increment in time between lines of the table, and the ending time, where all the time values must be less than 48 hours. After your table is printed out, also print the peak altitude and its corresponding time.

Specifications:
The format for your output table numbers should be to use 2 values after the decimal point for the time and velocity, but only 1 value after the decimal point for the altitude. To format your table, use the "`\t`" (tab) character. You must use a **`while`** loop to solve the problem correctly, since floating point numbers are being used.

HINT: You may use the math library and the **`pow()`** function if you want to. For example, to calculate t^4, use **`pow(t,4)`**, assuming that your variable for time is called **`t`**. Be sure to use include the math library in your preprocessor directives.

As an example, if you execute the program with the following underlined inputs, the output will be:

```
~> hw11.o
Welcome to the Weather Balloon Altitude and Velocity Program.

Enter the Balloon's Starting Time: 1
Enter the Time Increment: 2.2
Enter the Balloon's Ending Time: 10
```

```
Weather Balloon Information

Time (hr)       Altitude (m)    Velocity (m/s)

  1.00            3951.9             0.94
  3.20            9829.4             0.56
  5.40           13066.7             0.27
  7.60           14298.6             0.05
  9.80           14092.3            -0.10

Peak Altitude: 14298.6 (m)
Corresponding Time: 7.60 hr
~>
```

If your program does not follow these specifications properly, the Grader Program will not accept your submission. Develop your I/O diagram and pseudocode, debug your code, and submit to the Grader Program.