**ENED 1090: Engineering Models I**

**Homework Assignment #6**

**Due: Week of October 26th at the beginning of your Recitation Section**

**Problem 1:** Consider the code show below. Complete the table showing the values for the variables at the end of each iteration through the loop. You may not need all the rows in the table below. **Do this without MATLAB** first then check your results using MATLAB.

**Total = 0;**

**Count = 0;**

**while abs(2-Total) > 0.1**

**Total = Total + 1/2^Count;**

**Count = Count + 1;**

**end**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **While Condition True (Y/N)?** | **Total** | **Count** |
| **Before Loop** |  | **0** | **0** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Problem 2:** In Homework #4 (Problem 2), you wrote a script file that prompted the user for Name, Age, City, State, and Zip Code. You asked the user if the information was correct and if the user answered No, they were given the opportunity to correct only one of the items entered. Modify the script (start with the script in the HW#4 solutions if you didn’t do this problem correctly on the previous assignment) to incorporate a **while loop** so that the user can make as many corrections to the information as he/she needs.

**TEST CASE #1: Run your script and when asked if the information is correct, answer no. When asked which item is incorrect, pick City and enter a different (or corrected) City. When asked a second time if the information is correct, answer yes.**

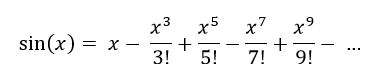
**Paste the Resulting Output here:**

**TEST CASE #2: Run your script and when asked if the information is correct, answer no. When asked which item is incorrect, pick Zip Code and enter a different Zip Code. When asked a second time if the information is correct, answer no again and indicate that the State is wrong. Correct the State then indicate that all information is now correct.**

**Paste the Resulting Output here:**

**PASTE SCRIPT HERE:**

**Problem 3:** In Homework #5 (Problem 3), you wrote a script file for the Taylor Series for the sine of an angle:



Modify the script (you can start with the script in HW#5 solutions if needed) to do the following:

* Prompt the user for an angle (in radians) between 0 and 2pi.
* **Use a** **while loop** to calculate the estimate for the sine of the angle by adding one additional term from the Taylor Series polynomial each iteration through the loop. The while loop should continue as long as the absolute value of the difference between the most recent estimate and the previous estimate exceeds 0.00001 (1e-5).
* Use fprintf statement(s) to display the estimate of the sine, the actual sine, and the number of terms of the Taylor Series polynomial required to achieve the specified accuracy. The actual and estimated values should be displayed using 6 places behind the decimal point. The number of terms should be displayed as an integer.

Run your script to complete the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Angle** | **Estimate of sin(Angle)** | **Actual sin(Angle)** | **Number of Terms** |
| **0** |  |  |  |
| **pi/8** |  |  |  |
| **pi/4** |  |  |  |
| **pi/2** |  |  |  |
| **pi** |  |  |  |
| **3\*pi/2** |  |  |  |
| **2\*pi** |  |  |  |

**PASTE SCRIPT HERE:**

**Problem 4:** In Homework #4 (Problem 4), you wrote a script file that allowed the user to play a single round of the dice game: Under-Over 7. Modify the script (or start with the script in the HW#4 Solutions) to add the following features:

* At the very beginning, ask the user how much money he/she has in total (balance) to bet during the course of the game.
* Add in code so the user cannot ever bet more money than he/she has. If the user attempts to bet more money than his/her balance, display the user’s balance and prompt them to enter a bet that doesn’t exceed their balance.
* Your script already displays to the user whether he/she won or lost and the amount won. Add in code to keep track of the user’s total amount of money (balance). Display the current balance to the user during each round of play.
* At the end of each round, the user should be asked if he/she wants to play again. Game play should only continue as long as the user wants to play again and the user still has money left.
* When the user runs out of money or the user indicates he/she no longer wants to play, the game should stop and the user’s final balance (which could be $0) should be displayed.

**Run your script several times to test it. Paste a sample output here that includes three rounds of play:**

**PASTE SCRIPT HERE:**