ENGR 102 – PRACTICE PROBLEMS

Spring 2019

These problems may or may not be similar to exam problems. However, working through these exercises should improve your program problem-solving, and assist you in doing well on the exam. I suggest first attempting these problems with no computer, calculator, cell-phone or other notes nearby. Once you've done what you can, then start utilizing other tools.

PRACTICE PROBLEM 01:

The Maclaurin series expansion for $\frac{1}{1-x}$ on an interval from $-1 < x < 1$ is as follows:

$$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + x^4 + \cdots + x^n$$

Write Python code which asks for input of a value of x on the interval $-1 < x < 1$, and which computes an approximation to $\frac{1}{1-x}$ using the using the series expansion summation. The summation should be continued until the term to be added to the summation is less than 10^{-6} in absolute value. Hint: Note that each term in the series is x raised to a power, including the 1 and x terms: $x^0 = 1 \ and\ x^1 = x$

PRACTICE PROBLEM 02:

Write a Python program that takes as inputs 5 integers. The program should check to see if any of the 5 are duplicates of another (i.e., check whether any of the integers were entered more than once). If, after all inputs are entered, a duplicate is found, the program should print "Duplicates", otherwise it should print "All Unique"

PRACTICE PROBLEM 03:

Write a program that will ask a user to enter names and ages of people, stopping when an age of 0 is entered (and not processing that person). The program should collect this information, and then output the average age, the name of the oldest person, and the name of the youngest person. Assume no two people have the same age.

PRACTICE PROBLEM 04:

Write a program that will ask the user to input two integers and calculate the sum of numbers between them that are multiples of 4. Exit the program if the user provides invalid input. Implement a loop to compute the partial sum. Do not use lists, tuples, arrays.

PRACTICE PROBLEM 05:

Write the Python code that is needed to print the numbers between 1 and 50 that are not divisible by 2 or by 3.

PRACTICE PROBLEM 06:

The function

$$P(x) = \frac{1}{\sqrt{2\,\pi s^2}} \exp\left(-\frac{(x-u)^2}{2s^2}\right)$$

is very important in probability and statistics. Write a user-defined function that takes as input parameters x, u, and s, returns the value of the function evaluated at these values. You can use the numpy function numpy.exp() to calculate the exponential.

Then, write the programming commands which call this function and plot it for x between -3 and 3 with $u = 0$ and $s = 1$ (for all values of x). Be sure to label your axes and show the plot.

PRACTICE PROBLEM 07:

Other than answering that the Top-Down process starts at the top and the Bottom-Up process starts at the bottom, describe the differences in the approach to programming, using the Top-Down Design Process and the Bottom-Up Design process. List 2 advantages of each design process.

PRACTICE PROBLEM 08:

Assume a function called isprime is available for you to use in a module called ENGR102 which determines if a number is a prime number or not. The input parameter for isprime is a single integer and the output is either True or False. Write the Python program that would input two integers and then test only the odd numbers between and including these two numbers, to see if they are prime numbers or not using the isprime function. The output of your program should be a list of prime numbers found. If no prime numbers were found, then output should state that no prime numbers were found. Start your code with: from ENGR102 import isprime.

PRACTICE PROBLEM 09:

What does the following python code output?

```
def myfunc(x):
    a = 1
    print(x)

a = 5
myfunc(a)
print(a)
```

PRACTICE PROBLEM 10:

Write a function that will take as input a set of numbers, and which returns the mean and variance of the set of numbers. The input can be a list, tuple or numpy.array. For a set of numbers $x_1, x_2, \ldots, x_n$, the mean is defined as $m = \frac{1}{n}\sum_{i=1}^{n} x_i$ and the variance is defined as $S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - m)^2$. DO NOT use the numpy functions numpy.mean and numpy.var ; you must use a loop in your function.

Practice Problem 11:

A text file named 'data.dat' is stored on a computer's hard drive. In a text editor the file displays:

```
# Created on November 15, 2018
# Time, Temperature, Windspeed
# (min), (deg F), (knots)
0, 33.47, 1.27
5, 32.59, 1.95
10, 33.62, 0.76
15, 33.79, 1.12
```

Write a program that reads the contents of this file, assigns the header lines to a variable that is a list of strings, and assigns the data to a variable that is a floating point np.array that contains all of the data in the file. (The data should be in a 4 x 3 array.) Write this code using standard file I/O commands; do not use any csv reader that you may know of in some Python package.

Practice Problem 12:

What is the output of the following python program?

```
    data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]
    print(data[1][0][0])
```

Practice Problem 13:

Write a program that takes in coefficients for a polynomial equation and prints the complete equation. Write a function that takes the derivative of any polynomial equation and call that function to take the derivative of the inputted polynomial equation and print out the derivative.

Practice Problem 14:

Write a program that intakes an arbitrary number of patients' names and signs/symptoms; including blood pressure, pulse, respiratory rate, and blood glucose. Have the information assigned to the patient's name in a Python dictionary. Then print out the patient's information when the user types in that person's name to a prompt.

Practice Problem 15:

Create a program that will open up a file named 'grades.txt' and then find and print the average score, the maximum score and the minimum score. Each grade will be on a new line in the file. Do not use a .csv reader for this problem.

Practice Problem 16:

Create a function(numcondition()) which asks user two integer values, and returns true if both add up and multiply to more than 10 and 20 respectively, and false if neither of these conditions are satisfied. If one is satisfied, it returns which condition is satisfied. use test cases and docstring.

Practice Problem 17:

Write a function in python that asks for the ideal mass of products in a chemical reaction and then for the actual mass produced. The function should then return the percent yield of the chemical reaction. Then write another function that will take the return of the first function and add it to a .txt file. Each time the function is run, the next percent yield should be documented on another line.

Practice Problem 18:

Create a program that makes two warriors fight each other. Each warrior should have a critical strike chance, an armor stat (that reduces damage taken proportionally), an attack range (minimum and maximum of possible damage to deal), an evasion stat, and a health stat that are all given by user input. Print the number of rounds (each round being an attack from one of the warriors) and the health of each warrior and the damage dealt after each round. (This program should use arrays, loops, and external libraries.)

Example Problem 19:

A vending machine will return the fewest number of coins possible as change. Take as input the amount of change required, and provide as output how many of each type of coin will be dispensed. Assume only dollar coins, quarters, dimes, nickels and pennies are available.

Example Problem 20:

Define a function that takes a word and counts the number of consonants & vowels. Use this function to perform the same action for each word in a sentence inputted by a user. (Treat “y” as a consonant, unless you want a challenge.)

Example Problem 21:

Write a function to insert a number to an ordered list (assume it goes from small to big) so that the list still goes from small to big. Return the updated list. Do not use a “sort” method