Prior Exam Questions

1. The function below 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.

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

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.

2. 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 approach to programming using the Top- Down Design process and the Bottom-Up Design process. List 2 advantages of each design process.

3. 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.

4. Write a Python program that can be used to find the x,y coordinates of the local maxima and the minima points between x = -3 and x=5 for the polynomial $y = x^3 - 3x^2 - 5.5x + 25$. The output should be the coordinates for both the maxima and the minima.

5. 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.

6. 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, …, x_n, the mean is defined as shown below. Do NOT use the numpy functions numpy.mean and numpy.var; you must use a loop in your function.

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - m)^2$$

7. Given two numpy arrays A and B, write the Python code to calculate C = AB (matrix multiplication (not by-element)). Assume that the inner dimensions of A and B match.

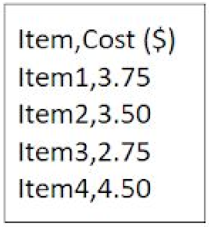
8. A *perfect number* is a positive integer >1 that is equal to the sum of its proper divisors. The smallest perfect number is 6 since the sum of the proper divisors for 6 (1, 2, and 3) equals 6. The integer 28 is also a perfect number since the divisors for 28 are 1, 2, 4, 7, and 14 and the sum of these numbers equals 28.

   Write a function that takes as input a positive integer >= 1, then determines the proper divisors of that integer, sums these divisors, and returns a True value if the number is a perfect number or a False value if it is not.

   Assume that the program that calls this function verifies that the input is a positive integer >= 1.

   Recall that "The proper divisors of a positive integer N are those numbers, other than N itself, that divide N without remainder. For N > 1 they will always include 1, but for N == 1 there are no proper divisors. The proper divisors of 6 are 1, 2, and 3. The proper divisors of 100 are 1, 2, 4, 5, 10, 20, 25, and 50."

9. A text file named `item_cost.dat` is stored on a computer's hard drive. In a text editor the file displays the first few lines of the file (note that the first line contains a header).



```
Item,Cost ($)
Item1,3.75
Item2,3.50
Item3,2.75
Item4,4.50
```

   Write a program that reads the contents of this file, determines the cheapest item and prints the information about that item and its cost to the screen using the following format:

```
The cheapest item is Item57. It costs $1.05.
```

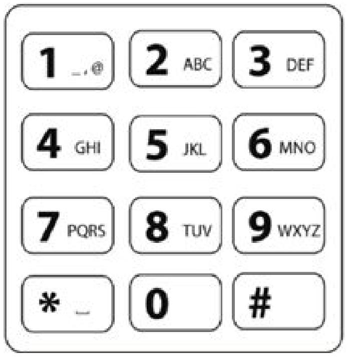
   Write this code using standard file I/O commands; *do not use* any csv reader available in some Python package. You cannot use any built-in sorting functions. You should use a list (or more than one list), a loop (or more than one loop). You may assume that all costs are unique (no two items cost the same).

10. Explain what happens (what will be output) when the code below is executed. If the code generates and error, explain the cause of the error.

```
print(int(‘5.5’))
```

11. Given the string A='Aggie Engineers Rock And Are In High Demand By Industry'. Write the Python code to prepare a list of the words in this string.

12. A schematic for converting Phone letters to digits mapping is shown in the image on the right. Write a Python program that prompts the user to enter a 10-character phone number in this format XXX-XXXXXXX. The program replaces the last seven alphabetic characters by their equivalent digits and displays the entered phone number in this format XXX-XXX-XXXX. For example, if 800-GOFEDEX is entered, the program displays '800-GOFEDEX is equivalent to 800-463-3339'. You may assume that the last seven characters are alphabetic characters from A – Z.



13. Write a program that will input data (namely, the results for a recent ENGR 102 quiz) from the keyboard, save it to a file in the same directory where your python files are saved, and then print the average quiz grade to the screen. Your program should prompt the user for the name of the file to create, how many data entries there are, and then request the values be entered until the user entries are complete. Each entry includes a student's name and his/her score. Write this code using standard file I/O commands. Before the program terminates, print the average quiz grade to the screen with the appropriate format:

```
The average grade for this quiz is 78.2
```

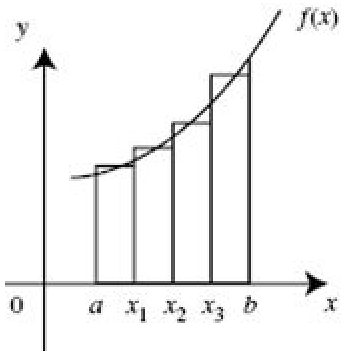
When opened in a text editor, the output file should display the following (for example) and be nicely formatted (note the header line and the alignment of columns):

```
Name         Score
Tom          85
Jim          96.5
Helen        92
Sam          89.5
Tina         97
```

14. Given the equation $y = x^3 - 2.1\,x^2 + 14x - 5$, write the Python code that will integrate this function by calculating the area under the curve from one value to another. Your program should first prompt the user to input values for the integration limits ***a*** and ***b*** and check to see if ***b*** > **a**. If this is not true, ask the user to input these limits again, and continue to check until this condition is satisfied. Next prepare a function called `area_calculator` that takes as input the two limits ***a*** and ***b***, and returns the area under the function for the equation shown above. Your function should integrate this equation by breaking the area between these two limits into 200 elements, and then sum the areas of each element. For this problem, the area of the element is defined as the product of the width of the element times the height at the mid point of the interval. The figure to the right is an attempt to show how the area under the curve is to be broken up into rectangular elements and the areas summed. After returning the value for the area, the main program should display a formatted output similar to:

```
The area between 2.12 and 6.45 is 312.04
```

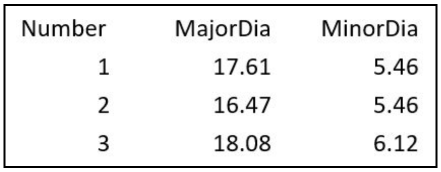
Note: consider this a 'challenge' activity; no calculus is covered this semester.



15. An Armstrong number is a positive 3 digit integer, where the sum of the cubes of its digits is equal to the integer itself. For example, 371 is an Armstrong number as (3**3)+ (7**3) + (1**3) = 371

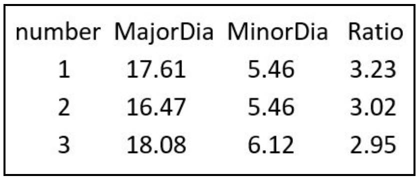
    Write a function called `Armstrong_number` that takes as input two positive 3 digit integers and returns a list or array or tuple containing all the Armstrong numbers between and including these two integers, and a True or False Boolean variable, True meaning an Armstrong value was found in the range. Assume that the code in the calling function is set to ensure that the second integer input to the function is greater than the first and that both values for *x* are in the range *100<=x<= 999*.

16. A text file named ‘`peanut.dat`’ of unknown length is stored on a computer's hard drive. In a text editor the file displays the first few lines of the file (note that the first line contains a header):



```
Number     MajorDia    MinorDia
     1        17.61        5.46
     2        16.47        5.46
     3        18.08        6.12
```

    Write a **program** that reads the contents of this file. Then for each peanut number, output to the screen the peanut number, Major Diameter, Minor Diameter and the diameter ratio in a formatted and ordered output, with headings, and the correct number of decimal places such as shown below for the first 3 peanuts:



```
number  MajorDia  MinorDia  Ratio
   1     17.61      5.46     3.23
   2     16.47      5.46     3.02
   3     18.08      6.12     2.95
```

    Write this code using standard file I/O commands; *do not use* any Python module csv reader. Use formatted output statements, reporting values to 2 decimal places; you may not use the round() function.

17. Write a program that will repeatedly ask the user to input a student's sex and age. The program will continue to ask for input until a negative age value is entered, meaning that the user is done inputting data. The program should determine the number of female students and their minimum and maximum ages, and the number of male students and their minimum and maximum ages. The results should be written to a new file using the name ‘Student_*Data_19Sp.txt*’. The file, when is opened in a text editor, should look like this:

```
Gender     Number of Students   Minimum Age   Maximum Age
Female     32                   17            24
Male       57                   17            26
```

    Note the required header line, data spacing, and column alignment

18. The Taylor series to approximate the cos (x) for all values of x, is defined as

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!}\, x^{2n}$$

Write a function called `Tay_term` that inputs a value of *n* and returns the value of the series for only that value of *n*, as shown in the formula above. For example, if a value of *n=3* was transferred to the function, and x = 0.5, `Tay_term` would return the value for $\frac{(-1)^3}{(2*3)!}\, x^{(2*3)}$ for x=0.5. The main program operates by first designating *x* as a global variable, then asking the user to input a value of *x*. To calculate the value for cosine of x using this series, the function `Tay_term` is repeated called with each increment of *n*, the values for each term are summed, until the value of the absolute value of the current term returned by the function is less than 0.00001. The output should display the results such as

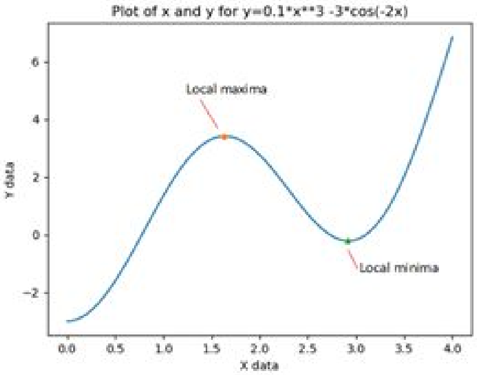
```
The cosine of 0.50 is 0.8776
```

Note: factorial() is a function available in the math module

19. Write a Python program that can be used to graph the following equation for *x* between *x=0* and *x=4*. Be certain to include a sufficient number of values of x so that a smooth curve can be obtained with your plot. Include a graph title and labels for the *x* and *y* axes.

y = 0.1x^3 - 3cos(-2x)

Next write a function that is to be called by your program to find the local maxima and minima points. Your function should import your, lists, arrays, or tuple of both the *x* and *y* data points and return the coordinates of the local maxima and minima. Plot the maxima/minima points on the graph and output/print the x,y coordinates for these two data points.


Plot of x and y for y=0.1*x**3 -3*cos(-2x)
Local maxima
Local minima
Y data
X data