Python basic assignment

*Note: Submit jupyter notebook file with its output shown there*

1. Write a program that repeatedly prompts a user for integer numbers until the user enters 'done'. Once 'done' is entered, print out the largest and smallest of the numbers. If the user enters anything other than a valid number, catch it with a try/except and put out an appropriate message and ignore the number.
2. Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error. If the score is between 0.0 and 1.0, print a grade using the following table:

Score Grade

>= 0.9 A

>= 0.8 B

>= 0.7 C

>= 0.6 D

< 0.6 F

If the user enters a value out of range, print a suitable error message and exit.

1. Write a program to prompt the user for hours and rate per hour using input to compute gross pay. Pay should be the normal rate for hours up to 40 and time-and-a-half for the hourly rate for all hours worked above 40 hours. Put the logic to do the computation of pay in a function called computepay() and use the function to do the computation. The function should return a value. Use 45 hours and a rate of 10.50 per hour to test the program (the pay should be 498.75). You should use input to read a string and float() to convert the string to a number.
2. Open the file romeo.txt and read it line by line. For each line, split the line into a list of words using the split() method. The program should build a list of words. For each word on each line check to see if the word is already in the list and if not append it to the list. When the program completes, sort and print the resulting words in alphabetical order.

*You can download the sample data at* [*http://www.py4e.com/code3/romeo.txt*](http://www.py4e.com/code3/romeo.txt)

1. Write a program that prompts for a file name, then opens that file and reads through the file, looking for lines of the form:

X-DSPAM-Confidence: 0.8475

Count these lines and extract the floating point values from each of the lines and compute the average of those values and produce an output like:

**Average spam confidence: 0.765**

*Do not use the sum() function or a variable named sum in your solution.*

*You can download the sample data at http://www.py4e.com/code3/mbox-short.txt*

1. Open the file mbox-short.txt and read it line by line. When you find a line that starts with 'From ' like the following line:

**From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008**

You will parse the From line using split() and print out the second word in the line (i.e. the entire address of the person who sent the message). Then print out a count at the end.

*Hint: make sure not to include the lines that start with 'From:'. Also look at the last line of the sample output to see how to print the count.*

*You can download the sample data at* [*http://www.py4e.com/code3/mbox-short.txt*](http://www.py4e.com/code3/mbox-short.txt)

1. Make a guessing game with necessary hints and exception handling.