DATA 200 Spring 2020 Homework 1

Due February 14, 2020 at 11:59pm

Note: Please only use standard python covered in class lectures. Python libraries such as pandas, numpy are *not* allowed in your work.

1. Write a Python program that can simulate a simple "decimal extractor", using the console as the exclusive input and output device. The program should prompt the user "Enter a number: " for input. Each input to the calculator should be a number like 12.34 or 456. After each such input, you should output to the Python console the decimal portion of the input value. For instance, 12.34 should return 0.34 and 456 should return 0. After each input, the output value should be printed and should prompt the user for input again and should only end if the user enters the string **end** as input. (5 points)

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 message. 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

   Run the program repeatedly as shown above to test different values for input. Here's some sample input and output:

```
Enter score: 0.95
A

Enter score: perfect
Bad score

Enter score: 10.0
Bad score

Enter score: 0.75
C

Enter score: 0.5
F
```

(5 points)

3. Write a Python function to extract the usernames from a list of email addresses and return them in list. Use **string manipulations** and **list comprehension** in your function. Below is the skeleton of what you should have. (5 points)

```
# emails - list of email addresses (strings)
def extract_usernames_domains(emails):

< your code here >

return usernames

emails = ["apple@gmail.com", "orange@yahoo.com", "grape@abc.net"]
usernames = extra_usernames_domains(emails)

# usernames == ["apple", "orange", "grape"]
```

(5 points)

4. Starting with `l1 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`, use list comprehension to create the following lists: (10 points)
   a. `[5, 6, 7, 8, 9, 10, 11, 12, 13, 14]`
   b. `[1, 3, 5, 7, 9]`
   c. `[1, 9, 81, 729, 6561]`
   d. `[0, 3, 4, 9, 8, 15, 12, 21, 16, 27]`
   e. `[1, 3, 10, 14, 18]`