CS5001: Lab 10. Due on Friday, Nov-17-2023.

Name(s): Xujia Qin

Email(s): qin.xuj@northeastern.edu

You can work on this lab either individually or in small group of two or three students. If working in a group, include names of all the students in the submission PDF.

Getting credit for this lab. This lab handout has several empty boxes that prompt you to answer a question. As part of the lab, you are to write the answers to these questions inside the boxes/blanks. When you are finished, you should create a PDF and upload it on Canvas. If you don't finish, you have until 11:59 PM on Friday, Nov-17-2023 to submit.

What computer to use? If your primary computer is a laptop, bring it to the lab to work on, as lab is an excellent opportunity to get started with Python on your machine. You should follow the instructions on the course website. Ask a TA for help if you have problems with your installation. If you prefer, you could also use one of the machines in the lab room to work on this lab assignment.

Lab Materials. Lab materials can always be found on Canvas under the appropriate lab posting.

For today's lab, you need: this handout, power.py, countVowels.py, recursiveMatch.py, sum_digits.py. All the files are posted on Canvas with the lab handout.

1 Exponent (20 points)

Complete the implementation of the following function in module power.py. The function power(base, exp) should return the value $base^{exp}$.

```
def power(base, exp):
    """ Returns an int equal to  base**exp
    Recursive implementation
    PreC: base, exp are nonnegative ints.
    """
```

2. Counting vowels (20 points)

Take a look at the module countVowels.py. Complete the implementation of the function countVowels. Your implementation should be recursive (use recursion). Your implementation should count the vowels in both lower case as well as upper case.

```
def countVowels(s):
    """ Returns the number of vowels in input s
    Recursive implementation
    PreC: s is a string.
    """
```

3 Recursive match (30 points)

Refer to the module recursiveMatch.py. Give an implementation of the following function so that it performs as specified. Your implementation should be recursive (use recursion). Your implementation should work for any two lists of integers, not just the two given in the script. Please test your code thoroughly by varying the two lists: **(24 points)**

For example, recursiveMatch([4, 2, 1, 6], [4, 3, 7, 6]) should return 2.

```
def recursiveMatch(l1, l2):
        """ returns the number of indexes where lists
        l1 has the same value as list l2
```

```
    Recursive implementation

    PreC: l1, l2 are lists of integers and have equal length.
    """
```

4. Sum of digits (30 points)

Refer to the module sum_digits.py and complete the implementation of the following function to do as directed in the docstring. The function should return the sum of digits of the number passed. E.g., for n=99, sum_digits() should return 9+9 = 18.

```
def sum_digits(n):
    """ Returns the sum of the digits of n

    Recursive implementation

    PreC: n is a positive integer.
    """
```

What to submit?

Submit the completed versions of the files power.py, countVowels.py, recursiveMatch.py, sum_digits.py that include your code for the functionality specified above in a single zip file.