## COMP 361/651 Data Analytics in Python/Data Analytics Techniques

8 possible points

**1** (2 pts.). Write code that keeps a running sum of numbers that are prompted for at the console and input until the user enters a 0, then output the sum. (Note that 0 counts a **`False`**.) Use a **`while`** loop. The following is an example execution.

```
Enter a number: 3
Enter a number: 2
Enter a number: -1
Enter a number: 4
Enter a number: 0
8
```

**2** (3 pts.). A palindrome is a word that reads the same forwards and backwards, such as 'civic', 'racecar, or 'Hannah'. One way to construct strings of letters that read the same forwards and backwards (which I hesitate to call palindromes since most are not even words) is to start with the empty string, **`''`**, and repeatedly add the same letter to the front and back, for example: **`''`** ➔ **`'nn'`** ➔ **`'anna'`** ➔ **`'hannah'`**. (To get a string of odd length, you start with a one-letter string.) Write Python code to produce (pseudo-)palindromes in this way. Have it so that it produces only palindromes of even length. (So, start with the empty string.) Use a **`while`** loop. Prompt for and input a letter at a time (which goes at both ends of the string generated so far). Enter an empty string (which counts as **`False`**) to stop.

**3** (3 pts.). Write code that prompts for and reads a list of integers. It splits the list in half and outputs the half with the larger sum. If the list is of odd length, it includes one more number in the second half then in the first half. The following are example runs.

```
Enter a list of numbers: [1, 3, 5, 6, 4, 2]
The larger half is  [6, 4, 2]
Enter a list of numbers: [6, 5, 3, 4, 1, 2]
The larger half is  [6, 5, 3]
Enter a list of numbers: [2, 3, 2, 4, 2]
The larger half is  [2, 4, 2]
```

Regarding the implementation, note function **`sum()`**, passed a list of numbers, returns the sum of the number. Also note that the **`eval()`** function evaluates a string containing a list of numbers into exactly that string of numbers. For example, **`eval('[1, 2, 3]')`** is the list of numbers **`[1, 2, 3]`**. This means that when we execute code like

```
lst = eval(input('Enter a list of numbers'))
```

and the user enters, say

```
[1, 2, 3]
```

**`lst`** is set to the list of numbers **`[1, 2, 3]`**.

To split the list of numbers in the middle, recall that, where **`lst`** is a list, **`len(lst)`** is its length. The operator **`//`** (two slashes) is integer division: where **`x`** and **`y`** are integers, **`x // y`** is the integer that is the truncated quotient of **`x`** by **`y`**. For example, **`9 // 2`** is **`4`**. So, use **`len()`** and **`//`** in finding where to split the list.