

CSC 241- Lab 3 (10 pts) – Submit a single file lab3.py containing solutions to the following problems:

When you have completed the function, run some tests in the IDLE shell. Copy and paste these into after your solution and comment out using a multiline comment (triple quotes). Submit the file with functions and commented out test runs.

1. Define a function **print3Chars** which accepts a single argument, a list of strings, and that **prints** to the screen, one per line, the first three characters of each string. If the string has fewer than three characters, it prints as many characters as the string has. You are allowed to index into a string, i.e. `s[2]`, but you are NOT ALLOWED to use slice notation, e.g. `s[1:5]`. Sample usage:

```
>>> print3Chars(['hello', 'goodbye', 'no', '', 'later'])
hel
goo
no

lat
>>> print3Chars(['a', 'b'])
a
b
>>>
```

2. Implement the function `avgLst()` that takes no parameters and inputs from the user a list of numbers. The function **returns** the average of the numbers in the list. If the list entered by the user is empty, the average returned should be `"n/a"`. You may assume that the list provided by the user contains only numbers, either floating point or integer or both. (Hint: Use the built-in sum function which works on lists: `sum([2,3,4])` is equal to 9). Sample usage:

```
>>> avgLst()
Please enter a list: [3,7,22]
10.666666666666666
>>> [3,7,22]
[3, 7, 22]
>>> avgLst()
Please enter a list: [-12.3,77.2,99]
54.63333333333333
>>> avgLst()
Please enter a list: []
n/a
>>>
```

3. Write a function `magnitude` that accepts a list of numbers and **returns** the magnitude of the list. The magnitude of the list is the largest absolute value found among numbers in the list. (Hint: it is the absolute value of either the largest or the smallest number in the list. Use the functions `abs`, `max`, `min`.) If the list is empty, then the function should **return** `"n/a"`. Sample usage:

```
>>> magnitude( [3,4,11,22])
22
>>> magnitude( [3,-4,11,-22])
22
>>> magnitude( [])
'n/a'
>>>
```

4. Write a function `printevens` that accepts a list of integers and **prints** all even numbers on the list. If the list is empty, the message `"Sorry, that list is empty."` should be printed instead. Sample usage:

```
>>> printevens( [3,4,11,22])
4
22
>>> printevens( [3,5,11,19])
>>> printevens([])
Sorry, that list is empty.
```

5. Implement a function `printDivisors` which accepts a positive integer `n` and prints all the divisors of that number. A divisor is a number between 1 and `n` that divides the number with no remainder. You will need `range` and `%`. Sample usage:

```
>>> printDivisors(12)
1
2
3
4
6
12
>>> printDivisors(17)
1
17
```