Simple list problems

- How would we write a function to convert a number from 1-12 into the corresponding month of the year as a string?

```python
def getmonth(month):
    #function code here
```

Examples of Concatenation

```python
a = [1, 2, 3]
b = [4, 5, 6]
c = a + b
print(c)  # prints [1, 2, 3, 4, 5, 6]

mylist = ["a", "b", "c"]
other = ["d", "e", "f"]
print(mylist + other)  #"a", 'b', 'c', 'd', 'e', 'f"]
```
Simple list problems

- What does this code do?

```python
lst = [2] * 3
lst2 = [4] * 2
lst3 = lst + lst2
for x in range(0, len(lst3), 2):
    lst3[x] = -1
```

Examples of List Slices

```python
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
numbers[2:]  # [3, 4, 5, 6, 7, 8, 9, 10]
nnumbers[:2]  # [1, 2, 3, 4, 5, 6, 7, 8]
nnumbers[1:8:2]  # [2, 4, 6, 8]
nnumbers[5::-1]  # [6, 5, 4, 3, 2, 1]
nnumbers[::-1]  # [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

Program Output

```
The total of the elements is 30
```

Using the repetition operator to initialize list

Program Output (with input shown in bold)

```
Enter the sales for each day.
Day #1: $1000
Day #2: $2000
Day #3: $3000
Day #4: $4000
Day #5: $5000
```

Here are the values you entered:

```
$1000.0
$2000.0
$3000.0
$4000.0
$5000.0
```

Program Output

```
```

11/3/2017
Practice

Get the file Nov1.py from my Box.com code directory. It has the main function written for you and stubs for 2 other functions that you will need to write.

findAverage(numbers) – will return the average of all the numbers in the list

countNumbers(numbers, average) - will return 2 values; it counts the number of above average and below average numbers in a list

Finding Items in Lists with the in Operator

• You can use the in operator to determine whether an item is contained in a list
  – General format: item in list
  – Returns True if the item is in the list, or False if it is not in the list

• Similarly you can use the not in operator to determine whether an item is not in a list

Example Using in Operator

```python
# This program demonstrates the in operator
# used with a list.
def main():
    # Create a list of product numbers.
    prod_nums = ['0415', 'r88?', 'q843', 'm688']
    # Get a product number to search for.
    search = input('Enter a product number: ')
    # Determine whether the product number is in the list.
    if search in prod_nums:
        print(search, 'was found in the list.')
    else:
        print(search, 'was not found in the list.')

# Call the main function.
main()
```

List Methods and Useful Built-in Functions

• append(item): used to add items to a list – item is appended to the end of the existing list

• index(item): used to determine where an item is located in a list
  – Returns the index of the first element in the list containing item
  – Raises ValueError exception if item not in the list
find() doesn’t exist for lists

- list_var.index(item)
- Searches left to right, returns position where found, but crashes if not found.
- Let’s build an algorithm that replicates find(), but works for lists (returns -1 if not found).

**List Methods and Useful Built-in Functions (cont’ed.)**

- **insert(index, item)**: used to insert item at position index in the list
- **sort()**: used to sort the elements of the list in ascending order
- **remove(item)**: removes the first occurrence of item in the list
- **reverse()**: reverses the order of the elements in the list

**Example Using Append**

```
def main():
    infile = open("randomNums.txt", 'r')
    numbers = []
    for line in infile:
        numbers.append(int(line))
    print(numbers)
main()
```

Output

```
[62, 57, 35, 27, 45, 44, 46, 68, 86, 27, 88, 33, 11, 61, 64, 45, 56, 9, 33, 32, 56, 63, 24, 26, 100, 95, 62, 10, 87, 58, 69, 54, 75, 43, 22, 93, 82, 16, 92, 49, 6, 71, 85, 59, 56, 22, 3, 50, 3, 20, 54, 18, 27, 78, 17, 43, 83, 38, 5, 64, 60, 92, 15, 26, 57, 39, 80, 41, 67, 56, 24, 77, 28, 90, 24, 72, 2, 46, 75, 53, 58, 47, 50, 18, 40, 65, 24, 58, 4, 58, 82, 40, 6, 77, 85, 86, 88, 63]
```

**Program 8.5**

```
# This program demonstrates the insert method.

def main():
    # Create a list with some names.
    names = ['James', 'Kathryn', 'Bill']
    # Display the list.
    print('The list before the insert:')
    print(names)
    # Insert a new name at element 0.
    names.insert(0, 'Joe')
    # Display the list again.
    print('The list after the insert:')
    print(names)
    # Call the main function.
    Main()

Program Output

The list before the insert:
['James', 'Kathryn', 'Bill']
The list after the insert:
['Joe', 'James', 'Kathryn', 'Bill']
```
List Methods and Useful Built-in Functions (cont’d.)

- **del statement**: removes an element from a specific index in a list
  - General format: `del list[i]`
- **min and max functions**: built-in functions that returns the item that has the lowest or highest value in a sequence
  - The sequence is passed as an argument
- **sum function**: built-in functions that returns the total of all the values in a sequence
  - The sequence is passed as an argument

Example Using `del`, `min`, `max`, and `sum` functions

Output

Before Deletion: `[5, 4, 3, 2, 50, 40, 30]`
After Deletion: `[5, 4, 3, 2, 50, 40, 30]`
The lowest value is 2
The highest value is 50
The sum of values in my list is 131
The lowest value is a
The highest value is d

Example Using del, min, max, and sum functions

```
my_list = [5, 4, 3, 2, 50, 40, 30]
del my_list[2]
print("After Deletion", my_list)
```

Output

Before Deletion: `[5, 4, 3, 2, 50, 40, 30]`
After Deletion: `[5, 4, 2, 50, 40, 30]`
The lowest value is 2
The highest value is 50
The sum of values in my list is 131
The lowest value is a
The highest value is d

Practice

Write a program that randomly generates 20 integers between 1 and 50, and stores them in a list. Print out the lowest and the highest numbers in your list, as well as the `sum` of all the numbers in the list.

Write a function that prints out sums of adjacent pairs of numbers in the list (don’t use sliding window; use indices)

```
print("The sum of values in my list is", round(sum(my_list)))
```

Output

Before Deletion: `[5, 4, 3, 2, 50, 40, 30]`
After Deletion: `[5, 4, 3, 2, 50, 40, 30]`
The lowest value is 2
The highest value is 50
The sum of values in my list is 131
The lowest value is a
The highest value is d

Write a function that takes a list and shifts all the elements in the list one spot to the left, without using slices! (the left-most element disappears)

```
shift_left(list):
    for i in range(len(list) - 1):
        list[i] = list[i+1]
    list[-1] = None
```

Example: `[1, 2, 3, 4, 5]` turns into `[2, 3, 4, 5, None]"