COP 2930 - Individual Programming Assignment #8

Due date: Please consult WebCourses for your due date/time

Objectives

1. To learn how to use lists.
2. To write functions that take in lists as parameters.
3. How to return a list from a function.
4. To integrate the working functions into a full program.

Problem A: Number List Program (numlist.py)

Write a program that will allow the user to add numbers to a list, remove numbers from a list and print the numbers in a list. At the beginning of the program, present the user with a menu with the following four options:

1. Add a number
2. Remove a number
3. Print the list
4. Quit

If the user chooses to add a number, ask them which number they want to add, and add it to the end of the list.

If the user chooses to remove a number, ask them which number they want to remove, check if that number is in the list. If it is not, print an error message stating that the number isn't in the list. If it is in the list, then remove the first occurrence of that number.

If the user chooses to print the list, do so.

After each selection, reprompt the user with the menu. The program ends when the user selects choice 4.

You may assume the user always properly enters information, so no error checking is necessary.

Input Specification

Note: It is guaranteed that whoever uses your program will adhere to these specifications. This means that you do NOT have to check for them!

Each menu choice entered will be an integer in between 1 and 4, inclusive.
Each number entered to add or remove will be an integer in between 1 and 100, inclusive.

Output Specification

For option 1, ask the question, "What number do you want to add?". After the user enters a number, print out, "Your number has been added."

For option 2, ask the question, "What number do you want to remove?"
If the number asked to be removed isn't in the list, just print out, "Sorry, that number is not in the list." If the number is in the list, remove it and then print out, "Your number has been removed."

For option 3, print the list in the default format that Python prints lists, with brackets denoting the start and end of the list and commas between items. Precede the printed list with the phrase, "Your list is."

After the user quits, print out, "Thanks for using the list generator!"

Output Samples

Below is one sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

Sample Run

```
What selection would you like to make?
1. Add a number
2. Remove a number
3. Print the list
4. Quit
1
What number do you want to add?
25
Your number has been added.

What selection would you like to make?
1. Add a number
2. Remove a number
3. Print the list
4. Quit
1
What number do you want to add?
16
Your number has been added.

What selection would you like to make?
1. Add a number
2. Remove a number
3. Print the list
4. Quit
3
Your list is [25, 16]
```

```
What selection would you like to make?
1. Add a number
2. Remove a number
3. Print the list
4. Quit
2
What number do you want to remove?
30
Sorry, that number is not on the list.

What selection would you like to make?
1. Add a number
2. Remove a number
3. Print the list
4. Quit
1
What number do you want to add?
30
Your number has been added.

What selection would you like to make?
1. Add a number
2. Remove a number
3. Print the list
4. Quit
2
What number do you want to remove?
16
Your number has been removed.

What selection would you like to make?
1. Add a number
2. Remove a number
3. Print the list
4. Quit
3
Your list is [25, 30]

What selection would you like to make?
1. Add a number
2. Remove a number
3. Print the list
4. Quit
4
Thank you for using the list generator!
```

Problem B: valsInRange Function (inrange.py)

Write a function that takes in a list of integers, **vals**, an integer **low**, and an integer **high** and returns the number of integers in **vals** that are in between **low** and **high**, inclusive. To make sure your function is working, test it using the test function given below:

```
def valsInRange(vals,low,high):
    # Fill in your code here.

def testValsInRange():
    print(valsInRange([3,12,6,5,2,8,9], 4, 9))
    print(valsInRange([100, 99, 98, 97], 0, 100))
    print(valsInRange([30,40,50,55,59,20], 60, 100))
    print(valsInRange([36,16,25,24,36,23,23,20,32], 23,35))

testValsInRange()
```

When you run the test, it should print 4, 4, 0, and 5, respectively, on separate lines.

Problem C: getValsInRange Function (listinrange.py)

Write a function that takes in a list of integers, **vals**, an integer **low**, and an integer **high** and returns a list with each value in **vals** that is in between **low** and **high**, inclusive.

```
def getValsInRange(vals,low,high):
    # Fill in your code here.

def testValsInRange():
    print(getValsInRange([3,12,6,5,2,8,9], 4, 9))
    print(getValsInRange([100, 99, 98, 97], 0, 100))
    print(getValsInRange([30,40,50,55,59,20], 60, 100))
    print(getValsInRange([36,16,25,24,36,23,23,20,32], 23,35))

testValsInRange()
```

When you run the test it should print the following:

```
[6, 5, 8, 9]
[100, 99, 98, 97]
[]
[25,24,23,23,32]
```

Restrictions

Please IDLE 3.6 (or higher) to develop your programs. Write each in a separate file with the names specified previously, **numlist.py**, **inrange.py,** and **listinrange.py**

Each of your **three** programs should include a header comment with the following information: your name, course number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details

Your programs will be graded upon the following criteria:

1) Your correctness and filling in the given function prototype appropriately.

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

3) Compatibility to IDLE.