Project 3- Working with Lists

Basic Information:

Project Heading:

Use the following as a header for all of your projects:

```
#-----------------------------------------------------------------------------------------------------------------
# Program name – filename.py
# Written by – your name
# Date – todays date
# Description of the program.
#-----------------------------------------------------------------------------------------------------------------
```

Style:

Follow the Python Style Guide available on Blackboard. **Be sure you have header comments for each of your functions.**

Due Date:

Week 11, 10/30/16

Turn in:

1. Algorithm or flow chart
2. Program listing for
    a. MyFavoritesPizzas_YourName.pl
    b. MoreGuests_YourName.pl

Problem Specification:

Chapters 3-4 of your text introduces lists. For this project you are to work the exercises:

1. 3-4 Guest List
2. 3-5 Changing Guest List
3. 3-6 More Guest List. In addition, use len() to print a message indicating the number of people you are inviting to dinner.
4. 4-1 Pizzas
5. 4-11 My Pizzas, Your Pizzas

Specific Requirements:

- Your program must use meaningful functions.
- Your program should not accept invalid choices from the player.
- Make sure that you do not reference global variables inside of your functions.

3-4. Guest List: If you could invite anyone, living or deceased, to dinner, who would you invite? Make a list that includes at least three people you'd like to invite to dinner. Then use your list to print a message to each person, inviting them to dinner.

3-5. Changing Guest List: You just heard that one of your guests can't make the dinner, so you need to send out a new set of invitations. You'll have to think of someone else to invite.

- Start with your program from Exercise 3-4. Add a `print` statement at the end of your program stating the name of the guest who can't make it.
- Modify your list, replacing the name of the guest who can't make it with the name of the new person you are inviting.
- Print a second set of invitation messages, one for each person who is still in your list.

3-6. More Guests: You just found a bigger dinner table, so now more space is available. Think of three more guests to invite to dinner.

- Start with your program from Exercise 3-4 or Exercise 3-5. Add a `print` statement to the end of your program informing people that you found a bigger dinner table.
- Use `insert()` to add one new guest to the beginning of your list.
- Use `insert()` to add one new guest to the middle of your list.
- Use `append()` to add one new guest to the end of your list.
- Print a new set of invitation messages, one for each person in your list.

4-1. Pizzas: Think of at least three kinds of your favorite pizza. Store these pizza names in a list, and then use a `for` loop to print the name of each pizza.

- Modify your `for` loop to print a sentence using the name of the pizza instead of printing just the name of the pizza. For each pizza you should have one line of output containing a simple statement like *I like pepperoni pizza*.
- Add a line at the end of your program, outside the `for` loop, that states how much you like pizza. The output should consist of three or more lines about the kinds of pizza you like and then an additional sentence, such as *I really love pizza!*

4-11. My Pizzas, Your Pizzas: Start with your program from Exercise 4-1 (page 60). Make a copy of the list of pizzas, and call it `friend_pizzas`. Then, do the following:

- Add a new pizza to the original list.
- Add a different pizza to the list `friend_pizzas`.
- Prove that you have two separate lists. Print the message, *My favorite pizzas are:*, and then use a `for` loop to print the first list. Print the message, *My friend's favorite pizzas are:*, and then use a `for` loop to print the second list. Make sure each new pizza is stored in the appropriate list.