Assignment 8 – Functions #2

Learning Objective

Define and call functions to simulate a game of Crack the Code.

Assignment Description

Write a program that generates a random 3-digit secret code. The user will try to crack the code by guessing digits that are in the code. While they have not guessed correctly, they will be given hints and allowed to guess new digits.

Your program will generate a list of single digit numbers. The user will then guess the digits. Hints will be given to the user such as how many times one of their digits occurs in the code and if each digit is in the correct position.

Steps

1. In PyCharm (Community Edition), open your existing ITP115 project.
2. Under the Assignments directory, create a new directory called **a8_*last_first*** where *last* is your last/family name and *first* is your preferred first name. Use all lowercase letters.
3. In the directory, create a new Python file called **assignment8.py**.
4. At the top of the file, put comments in the following format and replace the name, email, and section with your actual information:

```
# Name, USC email
# ITP 115, Fall 2022
# Section: number or nickname
# Assignment 8
# Description:
# Describe what this program does.
```

5. Import the random module.
6. Define the **isSingleDigit(userStr)** function.
   - Parameter: userStr is a string with input from the user

- Return value: a Boolean (True or False) depending if userStr is a string containing a one single digit
- Check to see if the userStr parameter contains only digit and the length is 1. If so, return True. If not, return False.
- You will call this function in the getUserList() function.

7. Define the **generateCodeList(size)** function.
   - Parameter: size is an integer which determines the length of the list
   - Return value: a list containing randomly selected integers between 0 and 9 (inclusive)
   - Create a variable that initially holds an empty list.
   - Use a loop depending on the size to randomly select an integer between 0 and 9 (inclusive) and add it to the list.
   - Return the list.
   - You will call this function in the main() function.
8. Define the **getUserList(size)** function.
   - Parameter: size is an integer which determines the length of the list
   - Return value: a list containing integers between 0 and 9 (inclusive) that the user entered
   - Display to the user the number of digits in the code using the size parameter. Here's an example when size is 3:

```
The number of digits in the code is 3
```

   - Create a variable that initially holds an empty list.
   - Use a range-based loop to repeat the number of times based on the size.
   - In the range-based loop, get input from the user with the position (index). Here's an example the first time in the loop:

```
Enter a digit at index 0:
```

   - Use a nested while loop, to continue repeating while the user's input (a string) is not a single digit. Call the isSingleDigit() function that you created earlier. Here is an example with the user's input in green:

```
Enter a digit at index 0: zero
Enter a digit at index 0:  1
Enter a digit at index 0: 22
Enter a digit at index 0: 0
```

- After the while loop, convert the user's input to an integer and add it to the list.
- After the range-based loop, display the list to the user. Here is an example:

```
Your guess is [0, 1, 2]
```

- Return the list.
- You will call this function in the main() function.

9. Define the **printHints(codeList, userList)** function.
   - Parameter 1: codeList is a list with the code that the user is trying to crack
   - Parameter 2: userList is a list with a guess of the code from the user
   - Return value: None
   - Print a message saying that you are generating hints:

```
Generating hints...
```

   - Create an int variable to be used to count the number of hints. You will need to update appropriately in this function. At the end of the function, if no hints were displayed, display the following to the user:

```
No correct digits
```

   - Loop through the userList parameter. In the loop, use the list.count() method to count the number of times each digit occurs. If the digit occurs, display a message to the user with the number of times. Here is an example when the digit 8 occurs 1 time in the list:

```
8 occurs 1 time(s) in the code
```

   - Now we want to see whether the user guessed a digit in the correct position. This loop is separate from the previous loop. Use a range-based loop depending on the size of one of the lists to get each index. Check the two lists to see if the digits are equal at that index. If so, display a message to the user stating that the digit at that index is in the correct position. Here is an example:

```
8 is in the correct position
```

- You will call this function in the main() function.

10. Define and call the **main()** function. This function will not have any parameters or a return value.
    - Create an int variable for the size of the secret code and set the value to 3.
    - Call the generateCodeList() function to generate a list containing random single digit integers. Use the size variable as the argument. Make sure to capture the function's return value in a variable.
    - Call the getUserList() function to generate a list containing single digit integers from the user. Use the size variable as the argument. Make sure to capture the function's return value in a variable.
    - Create an int variable to count the number of guesses it takes for the user to crack the code.
    - Use a loop to repeat while the code list does not equal the user's list.
    - In the loop, call the printHints() function using the appropriate variables as the arguments. Make sure to how the user guess again and increase the count of guesses.
    - After the loop, print the number of guesses it took the user to crack the code. Here is an example when the user took 7 guesses.

```
You cracked the code in 7 guesses!
```

    - After you defined the main() functions, don't forget to call it.
11. Be sure to comment your code. This means that there should be comments throughout your code. Put a comment block before each function stating the parameters, return values, and what that function does. Points will be deducted for not having comments.
12. Follow coding conventions. You should use lowerCamelCase or snake_case for variable names. You are welcome to create any variables that you need.
13. Test the program. Look at the Sample Output below. Assignments that do not run are subject to 20% penalty.
14. Prepare your submission:

- Find the **a8_*last_first*** folder on your computer and compress it. This cannot be done within PyCharm.
- On Windows, use **File Explorer** to select the folder. Right click and select the Send to -> Compressed (zipped) folder option. This will create a zip file.
- On Mac OS, use **Finder** to select the folder. Right click and select the Compress "*FolderName*" option. This will create a zip file.

15. Upload the zip file to your Blackboard section:
    - On Blackboard, navigate to the appropriate item.
    - Click on the specific item for this assignment.
    - Click on the **Browse Local Files** button and select the zip file.
    - Click the **Submit** button.

Grading

- This assignment is worth 40 points.
- Make sure that you the program runs. Points will be taken off if the graders have to edit the source code to test your program.
- Make sure to submit your assignment correctly as described above. Points will be taken off for improper submission.

Item	Points
isSingleDigit()	6
generateCodeList()	6
getUserList()	10
printHints()	10
main()	8
Total	**40**

Sample Output

Example #1

```
The number of digits in the code is 3
Enter a digit at index 0: zero
Enter a digit at index 0:  1
Enter a digit at index 0: 22
Enter a digit at index 0: 0
Enter a digit at index 1: one
Enter a digit at index 1: 10
Enter a digit at index 1: 1
Enter a digit at index 2: two
Enter a digit at index 2:  2
Enter a digit at index 2: 2
Your guess is [0, 1, 2]

Generating hints...
1 occurs 1 time(s) in the code

The number of digits in the code is 3
Enter a digit at index 0: 1
Enter a digit at index 1: 3
Enter a digit at index 2: 4
Your guess is [1, 3, 4]

Generating hints...
1 occurs 1 time(s) in the code
4 occurs 1 time(s) in the code

The number of digits in the code is 3
Enter a digit at index 0: 4
Enter a digit at index 1: 5
Enter a digit at index 2: 1
Your guess is [4, 5, 1]

Generating hints...
4 occurs 1 time(s) in the code
1 occurs 1 time(s) in the code
1 is in the correct position

The number of digits in the code is 3
Enter a digit at index 0: 6
Enter a digit at index 1: 4
Enter a digit at index 2: 1
Your guess is [6, 4, 1]
```

```
Generating hints...
4 occurs 1 time(s) in the code
1 occurs 1 time(s) in the code
4 is in the correct position
1 is in the correct position

The number of digits in the code is 3
Enter a digit at index 0: 7
Enter a digit at index 1: 4
Enter a digit at index 2: 1
Your guess is [7, 4, 1]

Generating hints...
4 occurs 1 time(s) in the code
1 occurs 1 time(s) in the code
4 is in the correct position
1 is in the correct position

The number of digits in the code is 3
Enter a digit at index 0: 8
Enter a digit at index 1: 4
Enter a digit at index 2: 1
Your guess is [8, 4, 1]

Generating hints...
4 occurs 1 time(s) in the code
1 occurs 1 time(s) in the code
4 is in the correct position
1 is in the correct position

The number of digits in the code is 3
Enter a digit at index 0: 9
Enter a digit at index 1: 4
Enter a digit at index 2: 1
Your guess is [9, 4, 1]

You cracked the code in 7 guesses!
```

Example #2

```
The number of digits in the code is 3
Enter a digit at index 0: 9
Enter a digit at index 1: 8
Enter a digit at index 2: 7
Your guess is [9, 8, 7]

Generating hints...
8 occurs 2 time(s) in the code

The number of digits in the code is 3
Enter a digit at index 0: 8
Enter a digit at index 1: 4
Enter a digit at index 2: 8
Your guess is [8, 4, 8]

Generating hints...
8 occurs 2 time(s) in the code
8 occurs 2 time(s) in the code
8 is in the correct position
8 is in the correct position

The number of digits in the code is 3
Enter a digit at index 0: 8
Enter a digit at index 1: 5
Enter a digit at index 2: 8
Your guess is [8, 5, 8]

Generating hints...
8 occurs 2 time(s) in the code
8 occurs 2 time(s) in the code
8 is in the correct position
8 is in the correct position

The number of digits in the code is 3
Enter a digit at index 0: 8
Enter a digit at index 1: 6
Enter a digit at index 2: 8
Your guess is [8, 6, 8]

You cracked the code in 4 guesses!
```

Example #3

```
The number of digits in the code is 3
Enter a digit at index 0: 5
Enter a digit at index 1: 6
Enter a digit at index 2: 7
Your guess is [5, 6, 7]

Generating hints...
No correct digits

The number of digits in the code is 3
Enter a digit at index 0: 4
Enter a digit at index 1: 3
Enter a digit at index 2: 2
Your guess is [4, 3, 2]

Generating hints...
4 occurs 1 time(s) in the code
2 occurs 1 time(s) in the code
4 is in the correct position

The number of digits in the code is 3
Enter a digit at index 0: 4
Enter a digit at index 1: 2
Enter a digit at index 2: 1
Your guess is [4, 2, 1]

Generating hints...
4 occurs 1 time(s) in the code
2 occurs 1 time(s) in the code
4 is in the correct position
2 is in the correct position

The number of digits in the code is 3
Enter a digit at index 0: 4
Enter a digit at index 1: 2
Enter a digit at index 2: 0
Your guess is [4, 2, 0]

You cracked the code in 4 guesses!
```

Example #4

```
The number of digits in the code is 3
Enter a digit at index 0: 1
Enter a digit at index 1: 2
Enter a digit at index 2: 3
Your guess is [1, 2, 3]

Generating hints...
1 occurs 1 time(s) in the code
2 occurs 2 time(s) in the code

The number of digits in the code is 3
Enter a digit at index 0: 2
Enter a digit at index 1: 1
Enter a digit at index 2: 2
Your guess is [2, 1, 2]

You cracked the code in 2 guesses!
```