Contestant Number: ____________________

Time: ___________________

Rank: _______________

PYTHON PROGRAMMING

(355)

REGIONAL 2023

APPLICATION KNOWLEDGE:

Combination Locks	_____________	(330 points)
TOTAL POINTS	_____________	***(330 points)***

Test Time: 60 minutes

GENERAL GUIDELINES:

Failure to adhere to any of the following rules will result in disqualification:

1. Contestant must hand in this test booklet and all printouts if any. Failure to do so will result in disqualification.
2. No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests (handwritten, photocopied, or keyed) are allowed in the testing area.
3. Electronic devices will be monitored according to ACT standards.

Combination Locks

People have always developed ways to keep items safe. The early days consisted of intricately tying ropes into knots. As time went on they developed locks made from wood and metal. Earliest known lock and key was used in Assyria followed by the Egyptian's wooden pin lock. Mechanical locks started over 6 thousand years ago in Egypt where a locksmith created a pin tumbler lock out of wood. The modern version of a lock that is still in use today was invented in 1848 by American Linus Yale Sr.

A local gym rents out locks for the lockers provided for the gym members. The type of lock that will be used includes several dials in the combination, but should have at least 3. Each dial has the options of 0 through 9. This allows over 10,000 combinations. The gym changes the combination before it is rented out. You will be creating a program that will use random numbers to create a set of possible combinations for this lock.

To use the application, the user will enter a digit to represent the number (at least 3) of individual dials (digits) used on the combination lock. Each of these individual digits will be between 0 and 9 (inclusive). The user will then need to enter how combinations will be generated. Then using pseudo random number generation, available using the random module, the program will display to the user a list of combination numbers.

Even though normal combinations for locks would not have duplicate numbers, for simplicity, we will allow for repeated numbers in our combination locks. This means that 2-9-7-9 would be an acceptable number, despite the 9 appearing twice.

Example output

```
This application generates possible lock combinations; assuming that the numbers will be from 0
to 9 (inclusive). Please first enter the quantity of dials on the combination locks. And then enter
how many combinations you would like to generate.


Each lock has how many dials? 3

How many combinations should I generate? 4

--------------------------------

Number 1: 3 7 8

Number 2: 6 5 3

Number 3: 8 4 9

Number 4: 9 0 8

--------------------------------

4 random combinations were generated
```

Requirements:

1. You must create a Python application named PP_355_ContestantNumber, where ContestantNumber is your BPA assigned contestant number (including dashes). For example, PP_355_01_2345_6789.
2. Your contestant number must appear as a comment at the top of the main source code file.
3. You will have four functions called *get_number_of_dials, get_how_many_to_list, get_number* and *next_combo_number*. You may have more functions.
4. Create a function named get_number_in_dials which accepts nothing (has no parameters) and prompts the user for the number of dials to generate in each lock combination. It needs to have at least 3 dials.
    a. If they enter 5, each lock combination will have 5 numbers: 7 5 3 9 2
    b. If they enter 3, each lock combination will have only 3 numbers: 9 4 7
    c. Check for that the number of dials is at least 3. After validity is reported, the user shall be prompted for an additional number to check
5. Create a function named get_how_many_to_list which accepts nothing (has no parameters) and prompts the user for the number of lock combinations to display.
6. Create a function named get_number which accepts two integers, one is the minimum and the second is the maximum value that could be returned. The function will return a random integer which falls within the minimum and maximum values specified (inclusive).
    a. You must use the randrange function.
    b. Do not use the randint function.
7. Create a function named next_combo_number which accepts an integer for how many individual numbers are in the lock combination and returns a string representation of a single combo number.
    a. You will need get a new random number for each individual number in the lock combination.
        i. Each number must fall between 0 and 9 (inclusive).
            1. You need to use the function which you wrote to get your next number.
        ii. Combine each formatted number together to make the final lock combination
8. For each lock combination you need to list to the user,
    a. Get the combo number and display it to the user in the format of:
        i. Number {order} {combo number}
        ii. See example above
9. Let the user know how many numbers were displayed.

Your application will be graded on the following criteria:

Solution and Project

The project is present on the flash drive	____ 10 pts
The project is named according to the naming conventions	____ 10 pts

Program Execution

Code copied to USB drive and the program runs from USB	____ 10 pts

If the program does not execute, then the remaining items in this section receive a score of zero.

Application asks the user of the number of dials and how many combinations to print	____ 20 pts
Application displays error message if invalid numerical input	____ 20 pts
Application successfully prints the correct number of combinations and numbers to represent the dials	____ 50 pts
Application lists the number of combinations generated.	____ 20 pts

Source Code Review

Code is commented at the top, for each function, and as needed	____ 20 pts
Code uses reasonable and consistent variable naming conventions	____ 20 pts
The **get_number_in_dials** function is present and is constructed in a logical manner	____ 30 pts
The **get_how_many_to_list** function is present and is constructed in a logical manner	____ 30 pts
The **get_number** function is present and is constructed in a logical manner	____ 40 pts
The **next_combo_number** function is present and is constructed in a logical manner	____ 50 pts

Total Points: ____ 330 pts