Lab exercise Week 5

You are asked to program a simple number guessing game. The computer will generated a secret number, with 3 digits that do not repeat, between 102 and 987. The game consists in guessing this 3 digit number in a maximum of 10 attempts.

Here is an example of execution.

```
I am thinking of a 3 digit number. Try to guess what it is. You have 10
attempts.
If you guess none of the digits I will answer WRONG
I will answer DING for every correct digit in the right position
and will answer DONG for every correct digit in the wrong position.
Enter a three digit number with non-repeating digits
Guess attempt 1:123
DING DONG
Enter a three digit number with non-repeating digits
Guess attempt 2:124
DING
Enter a three digit number with non-repeating digits
Guess attempt 3:245
WRONG
Enter a three digit number with non-repeating digits
Guess attempt 4:136
DING DING
Enter a three digit number with non-repeating digits
Guess attempt 5:137
DING DING
Enter a three digit number with non-repeating digits
Guess attempt 6:138
DING DING
Enter a three digit number with non-repeating digits
Guess attempt 7:139
You guessed it after 7 attempts. The number is indeed 139
Do you want to play again (Y/N)?
```

1. Write a python class ***Secret_Number***
    a. An instance of the class has the property ***number***, which is a string containing the 3 digit secret number. When an instance of the class is created the property ***number*** is initialized with a string containing a random number with 3 digits that do not repeat and does not start with 0.
    b. The class has a method *get(). get()* simply returns the property ***number***.
    c. The method also has the method getClues() which receives a guess as a string with 3 digits and returns the string “CORRECT” if the guess matches the property ***number***, “WRONG” if none of the digits match, or a string with as many “DING” as there are digits that match and in the correct position and as many DONG for digits that match but are not in the correct position.

For instance, here are examples of output for different guesses if the secret number is 123:

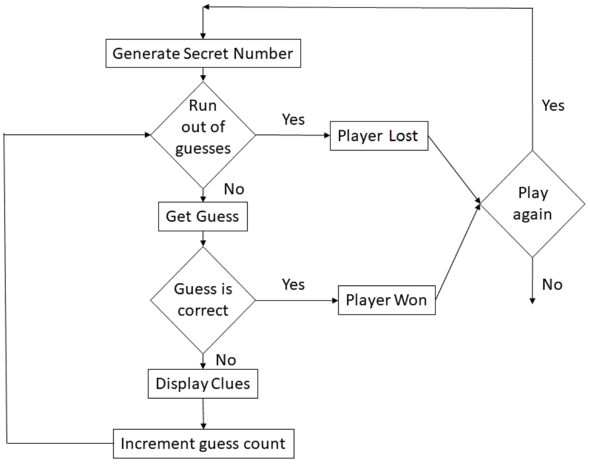
```
456  WRONG
729  DING
102  DING DONG
120  DING DING
132  DING DONG DONG
123  CORRECT
```

   d. Test your class in isolation.

2. Write a function *readNumber()* which will read a number with three digits as a guess from the user.
   a. The function will make sure the input is made of only digits, its length is exactly 3 and the digits are not repeated.
   b. The function will print "Enter a three digit number with non-repeating digits" and continue to prompt "Guess attempt %d" until an acceptable guess is input. %d is the attempt number which is provided as an argument to the function.
3. Write the program that creates an instance from the class **Secret_Number** and provides up to 10 attempts to the user to guess the secret 3 digit number. At each attempts, the program should display the DING DONG clues as per the method *getClues()* or "You guessed it after %d attempts. The number is indeed …" if the guess was right.

At the end, the computer should ask whether the user would like to play again, and repeat the process if the answer is positive.

A flowchart for the program could look like this:



Optional (1): Change your program so that the number of attempts 10 is not hard coded but you would ask the user who many attempts they would like to try for a given game session.

Optional (2): Change your program (including the class) such that the length of the secret number is not just hard coded to be 3 but can be parametrized by a constant in the program called NUM_DIGITS which can be assigned to be 3 or lower or higher.