Winning Ticket!

Your favorite uncle, Morty, is crazy about the lottery and even crazier about how he picks his “lucky” numbers. And even though his “never fail” strategy has yet to succeed, Uncle Morty doesn't let that get him down.

Every week he searches through the Sunday newspaper to find a string of digits that might be potential lottery picks. But this week the newspaper has moved to a new electronic format, and instead of a comfortable pile of papers, Uncle Morty receives a text file with the stories.

Help your Uncle find his lotto picks. Given a large series of number strings, return each that might be suitable for a lottery ticket pick.

Note that a valid lottery ticket must have 7 unique numbers between 1 and 59, digits must be used in order, and every digit must be used exactly once.

For example, given the following strings:

[ "569815571556", “4938532894754”, “1234567”, “472844278465445”]

Your function should return:

4938532894754 -> 49 38 53 28 9 47 54

1234567 -> 1 2 3 4 5 6 7

**SOLUTION**

The whole code is written in python. To run the program write ‘python Lottery.py’ in the CLI. I ran the program in PyCharm IDE.

The following is the code flow:-

* run\_program(input) 🡪
* find\_lottery(input) 🡪
* Check validity for each string in input using split\_num(input[i]) function 🡪
* In case of a lottery number greater than 7 digits in size, call singleDoubleValidity(num\_str) 🡪
* Find for pairs less than 59 and with repeating digits, e.g. 44, call findDoubleDigits(num\_str,pairCount).

The run\_program will print the valid numbers Uncle Morty can use in his next lottery, along with the 7 valid numbers used for each lottery number.

Code Efficiency:-

The time complexity of the code, to check the validity of a given string of Lottery number is O(n) where n is the length of a lottery number. I have made use of Hash-map (O(n) space complexity) to check if any of the 7 numbers are repeating in a lottery number string.