Assignment # 2

Deadline: 30th April, 2021 11:59 pm

Instructions:

- **Please do your own work; plagiarized code will result in zero marks.**
- **Do all these ask tasks in one python file named as your roll number followed by your section e.g. 20-13456-Assignment2_b.py**
- **Please add comments in your codes**
- **Late assignments will not be accepted.**
- **Do not forget to write DOC STRINGS.**

Task#1

Write a function that takes two strings as input and prints them together on one 50-character line, with the first string left-justified, the second string right-justified, and as many periods between the input strings as needed. For example:

```
>>>print_left_right( 'hello', 'world')

hello........................................world
```

Task#2

Write a function that takes two character strings as input and return true if and only if string1 is longer than the string2 and false otherwise.

Task#3

Write a function that prints true if string2 occurs in string1 and false otherwise.

Task#4

Write a program that prompts the user to provide a single character from the alphabet. Print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.

Task#5

Roman numbers. Write a program that converts a positive integer into the Roman number system. The Roman number system has digits

I	1
V	5
X	10
L	50
C	100
D	500
M	1,000

Numbers are formed according to the following rules:

a. Only numbers up to 3,999 are represented.

b. As in the decimal system, the thousands, hundreds, tens, and ones are expressed separately.

c. The numbers 1 to 9 are expressed as

I 1
II 2
III 3
IV 4
V 5
VI 6
VII 7
VIII 8
IX 9

As you can see, an I preceding a V or X is subtracted from the value, and you can never have more than three I's in a row.

d. Tens and hundreds are done the same way, except that the letters X, L, C and C, D, M are used instead of I, V, X, respectively.

Your program should take an input, such as 1978, and convert it to Roman numerals, MCMLXXVIII.

Task#6

Translate the following pseudocode for randomly permuting the characters in a string into a Python program.

```
Read a word.
Repeat len(word) times
    Pick a random position i in the word, but not the last position.
    Pick a random position j > i in the word.
    Swap the letters at positions j and i.
Print the word.
```

To swap the letters, construct substrings as follows:

first middle last i j

Then replace the string with

```
first + word[j] + middle + word[i] + last
```

Task#7

Write a program that reads an integer and displays, using asterisks, a filled and hollow square, placed next to each other. For example if the side length is 5, the program should display

```
*****   *****
*****   *   *
*****   *   *
*****   *   *
*****   *****
```

Task#8

The last digit of a credit card number is the check digit, which protects against transcription errors such as an error in a single digit or switching two digits. The following method is used to verify actual credit card numbers but, for simplicity, we will describe it for numbers with 8 digits instead of 16:

- Starting from the rightmost digit, form the sum of every other digit. For example, if the credit card number is 43589795, then you form the sum 5 + 7 + 8 + 3 = 23.

- Double each of the digits those were not included in the preceding step. Add all digits of the resulting numbers. For example, with the number given above, doubling the digits, starting with the next-to-last one, yields 18 18 10 8. Adding all digits in these values yields 1 + 8 + 1 + 8 + 1 + 0 + 8 = 27.

- Add the sums of the two preceding steps. If the last digit of the result is 0, the number is valid. In our case, 23 + 27 = 50, so the number is valid.

Write a program that implements this algorithm. The user should supply an 8-digit number, and you should print out whether the number is valid or not. If it is not valid, you should print out the value of the check digit that would make the number valid.

Task#9

Write a program that prompts the user for an integer and then prints out all prime numbers up to that integer. For example, when the user enters 20, the program should print

```
2
3
5
7
11
13
17
19
```