<div align="center">**PYTHON ASSIGNMENT (20 Marks)**</div>

Note: Submission date of Assignment is 20 May, 2020. Submission after 20th May will reduce 5 marks.

No one should have the same code, even not with different variable names, otherwise -1 will be done whose code seems to be same.

Submission procedure: Create a folder by your roll no. BT18CSEXXX <*XXX represents your roll number*>. Keep all the code in that folder with proper question no for each solution like "ques1.py", "ques2.py" also take the screenshot of your output and save it in the same folder like "output_ques1.jpg", "output_ques2.jpg" and so on.

1. Write a program that opens all *.txt* files in a folder and searches for any line that matches a user-supplied regular expression. The results should be printed to the screen.

2. You are given a file called class_scores.txt, where each line of the file contains a one-word username and a test score separated by spaces, like below:

   Rahul 83

   Rohan 86

   Write code that scans through the file, adds 5 points to each test score, and outputs the usernames and new test scores to a new file, scores2.txt.

3. You are given a file called students.txt. A typical line in the file looks like:

   rohan mishra   mishra123@gmail.com   555-3141

   There is a name, an email address, and a phone number, each separated by tabs. Write a program that reads through the file line-by-line, and for each line, capitalizes the first letter of the first and last name and adds the area code 301 to the phone number. Your program should write this to a new file called students2.txt. Here is what the first line of the new file should look like:

   Rohan Mishra   mishra123@gmail.com   301-555-3141

4. You are given a file called grades.txt, where each line of the file contains a one-word student username and three test scores separated by spaces, like below:.

   RAgrawal 83 77 54

   AMishra 86 69 90

   Write code that scans through the file and determines how many students passed all three tests.

5. Write a program that searches through all the files in a directory and prints the names of those files that contain the word 'python'.

6. You are given a file namelist.txt that contains a bunch of names. Some of the names are a first name and a last name separated by spaces, like George Washington, while others have a middle name, like John Quincy

Adams. There are no names consisting of just one word or more than three words. Write a program that asks the user to enter initials, like GW or JQA, and prints all the names that match those initials. Note that initials like JA should match both John Adams and John Quincy Adams.

7. Write a program that asks the user to enter a string. The program should create a new string called new_string from the user's string such that the second character is changed to an asterisk and three exclamation points are attached to the end of the string. Finally, print new_string. Typical output is shown below:

Enter your string: Alert

A*ert!!!

8. Write a program that asks 2 inputs from the user; i) to enter a word and ii) a character. The program should then print the following two lines: first line should be the part of the string including the entered character from the user, and second line should be the rest of the string after the character. Sample output is shown.

Enter a word: buffalo   Enter a character: a

buffa

lo

Note: If entered character is not present then output as **"no such character present in the given string"**

9. Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters abcde and ABCDE the program should print out AaBbCcDdEe.

10. How would you write a regex that matches a sentence where the first word is Alice, Bob, or Carol; the second word is either eats, pets, or throws; the third word is apples, cats, or baseballs; and the sentence ends with a period ?This regex should be case-insensitive. It must match the following:

- 'Alice eats apples.'
- 'Bob pets cats.'
- 'Carol throws baseballs.'
- 'Alice throws Apples.'
- 'BOB EATS CATS.'

but not the following:

- 'RoboCop eats apples.'
- 'ALICE THROWS FOOTBALLS.'
- 'Carol eats 7 cats.'

11. Write a function named printTable() that takes a list of lists of strings and displays it in a well-organized table with each column right-justified. Assume that all the inner lists will contain the same number of strings.

For example, the value could look like this:

tableData = [['apples', 'oranges', 'cherries', 'banana'],

['Alpha', 'Beta', 'Charlie', 'Delta'],

['dogs', 'cats', 'moose', 'goose']]

Your printTable() function would print the following:

apples Alpha dogs

oranges Beta cats

cherries Charlie moose

banana Delta goose

12. Write a program that takes a list of people's email addresses and a list of tasks that need to be done and randomly assigns unique tasks to people. Email each person their assigned tasks. For another possible feature, schedule the program to run once a week automatically.

Here's a hint: If you pass a list to the random.choice() function, it will return a randomly selected item from the list. Part of your code could look like this:

tasks = ['dishes', 'cooking', 'vacuum', 'walk dog']

randomTasks = random.choice(tasks)

tasks.remove(randomTasks) # this chore is now taken, so remove it.

13. Write a Python code to illustrate Sending mail from your Gmail account write comments in your code to explain each line. [2 M]

14. Create a python script to perform CURD operation on the database. [2 M]

15. Create a web page with heading, text area, and 2 buttons (submit and clear). After entering the text in the text area if user clicks the submit button, print the text area data in reverse order on the new web page. If user clicks clear button, then clear the text in the text area. [2 M]

16. Create a csv file by name "login.csv" which stores username and password. Create a web page that take username and password from the user. If user entered username and password are present in the csv file, redirect the user to the new page with welcome message else print the appropriate message. [2 M]