

## PC:503 Programming Lab Assignment – I

1. **Write a Python program from scratch to generate all permutations of a list in Python(without itertools).**

Input : [1,2,3]

Output : [(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)]

2. **Write a function that gives number of days of given year.**

Input : 1990

Input : 2044

Output : 365

Output : 366

3. **Print the name(s) of any car(s) having the second expensive car in. If there are multiple cars, order their names alphabetically and print each one on a new line.**

Input : [ ["BMW",5000000],  
["TOYOTA", 1000000],  
["NISSAN",1500000],  
["JAGUAR",8000000],  
["MERCEDES",8000000],  
["FERRARI",15000000]]

Output : JAGUAR  
MERCEDES

4. **Count the frequency of each character in a string and store it in a dictionary. An example is given below.**

Input: 'adcbddaacd'

Output: {'a': 3, 'b': 2, 'c': 2, 'd': 3}

5. **Output the space separated tuples of the cartesian product(without built-in function).**

Input : A = [1, 2]

B = [3, 4]

Output : Ax B = [(1, 3), (1, 4), (2, 3), (2, 4)]

6. **Write a Program which remove all the element with having count 2.**

Input : [1,2,3,4,4,5,5,5,6,7,6,8,8,8,8]

Output : [1,2,3,5,5,5,7,8,8,8,8]

7. **Write a program to find N largest and N smallest elements from a list. An example is given below.**

Input: N=3, Arr = [13, 29, 26, 15, 4, 70, 89, 57, 34, 66, 10, 49]

Output: Largest = [89, 70, 66]

Smallest = [4, 10, 13]

8. Write a program to reverse a stack without using reverse() function. An example is given below.

Input: stack = [3, 7, 2]

Output: stack = [2, 7, 3]

9. Write a function to print N “Perfect” numbers. A perfect number is a positive integer that is equal to the sum of its proper positive divisors. I.e. 6 is a perfect number because  $1+2+3 = 6$

Input: N = 4

Output: 6, 28, 496, 8128

10. Write a function to check whether the given password is valid or not. Password is a combination of alphanumeric characters along with special characters. A password is correct if and only if all the following conditions are satisfied:

- Length of the password should be a minimum of 10.
- The alphabet must be between [a-z].
- At least one numeric character [0-9].
- At least one alphabet should be of Upper Case [A-Z].
- At least 1 special character from [ \_ or @ or \$ ].
- No numeric character and special character should be adjacent.

An example is given below.

Input: D@iict1234

Output: Valid

Input: Daiict\_123

Output: Not Valid