***Python Task 2 Assignment***

**Q1. You are writing code for a company. The requirement of the company is that you create a python**

**function that will check whether the password entered by the user is correct or not. The function should**

**take the password as input and return the string “Valid Password” if the entered password follows the**

**below-given password guidelines else it should return “Invalid Password”.**

**Note: 1. The Password should contain at least two uppercase letters and at least two lowercase letters.**

**2. The Password should contain at least a number and three special characters.**

**3. The length of the password should be 10 characters long.**

**def check\_password(password):**

**# Check length**

**if len(password) != 10:**

**return "Invalid Password"**

**# Check uppercase letters**

**if sum(1 for c in password if c.isupper()) < 2:**

**return "Invalid Password"**

**# Check lowercase letters**

**if sum(1 for c in password if c.islower()) < 2:**

**return "Invalid Password"**

**# Check numbers**

**if sum(1 for c in password if c.isdigit()) < 1:**

**return "Invalid Password"**

**# Check special characters**

**special\_chars = "!@#$%^&\*()-\_=+[{]}\\|;:'\",<.>/?"**

**if sum(1 for c in password if c in special\_chars) < 3:**

**return "Invalid Password"**

**# If all checks pass, return "Valid Password"**

**return "Valid Password"**

**Example:**

**print(check\_password("MyPassw0rd!"))**

**# Output: Valid Password**

**print(check\_password("mybadpassword"))**

**# Output: Invalid Password**

**Q2. Solve the below-given questions using at least one of the following:**

**1. Lambda function**

**2. Filter function**

**3. Zap function**

**4. List Comprehension**

**B Check if the string starts with a particular letterY**

**B Check if the string is numericY**

**B Sort a list of tuples having fruit names and their quantity. [("mango",99),("orange",80), ("grapes", 1000)-**

**B Find the squares of numbers from 1 to 10Y**

**B Find the cube root of numbers from 1 to 10Y**

**B Check if a given number is evenY**

**B Filter odd numbers from the given list.**

**[1,2,3,4,5,6,7,8,9,10-**

**B Sort a list of integers into positive and negative integers lists.**

**[1,2,3,4,5,6,-1,-2,-3,-4,-5,0]**

**Using list comprehension:**

**strings = ['apple', 'banana', 'cherry', 'orange', 'kiwi']**

**letter = 'b'**

**result = [string for string in strings if string.startswith(letter)]**

**print(result)**

**# Output: ['banana']**

**Using filter function:**

**strings = ['apple', 'banana', 'cherry', 'orange', 'kiwi']**

**letter = 'b'**

**result = list(filter(lambda string: string.startswith(letter), strings))**

**print(result)**

**# Output: ['banana']**

**Check if the string is numeric:**

**Using list comprehension:**

**strings = ['apple', '123', '3.14', '45']**

**result = [string for string in strings if string.isnumeric()]**

**print(result)**

**# Output: ['123']**

**Using filter function:**

**strings = ['apple', '123', '3.14', '45']**

**result = list(filter(lambda string: string.isnumeric(), strings))**

**print(result)**

**# Output: ['123']**

**Sort a list of tuples having fruit names and their quantity:**

**Using lambda function:**

**fruits = [("mango",99),("orange",80), ("grapes", 1000)]**

**fruits\_sorted = sorted(fruits, key=lambda x: x[1])**

**print(fruits\_sorted)**

**# Output: [('orange', 80), ('mango', 99), ('grapes', 1000)]**

**Find the squares of numbers from 1 to 10:**

**Using list comprehension:**

**squares = [x\*\*2 for x in range(1, 11)]**

**print(squares)**

**# Output: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]**

**Using map function:**

**squares = list(map(lambda x: x\*\*2, range(1, 11)))**

**print(squares)**

**# Output: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]**

**Find the cube root of numbers from 1 to 10:**

**Using list comprehension:**

**cube\_roots = [round(x\*\*(1/3), 2) for x in range(1, 11)]**

**print(cube\_roots)**

**# Output: [1.0, 1.44, 1.71, 1.91, 2.08, 2.22, 2.35, 2.44, 2.54, 2.63]**

**Using map function:**

**import math**

**cube\_roots = list(map(lambda x: round(math.pow(x, 1/3), 2), range(1, 11)))**

**print(cube\_roots)**

**# Output: [1.0, 1.44, 1.71, 1.91, 2.08, 2.22, 2.35, 2.44, 2.54, 2.63]**

**Check if a given number is even:**

**Using lambda function:**

**is\_even = lambda x: x % 2 == 0**

**print(is\_even(4))**

**# Output: True**

**print(is\_even(5))**

**# Output: False**

**Using list comprehension:**

**numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]**

**even\_numbers = [x for x in numbers if x % 2 == 0]**

**print(even\_numbers**