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.

import re

def check\_password(password):

# Check password length

if len(password) != 10:

return "Invalid Password"

# Count uppercase and lowercase letters

uppercase\_count = sum(1 for c in password if c.isupper())

lowercase\_count = sum(1 for c in password if c.islower())

if uppercase\_count < 2 or lowercase\_count < 2:

return "Invalid Password"

# Check for at least one digit

if not any(c.isdigit() for c in password):

return "Invalid Password"

# Check for at least three special characters

special\_chars = re.findall(r'[!@#$%^&\*(),.?":{}|<>]', password)

if len(special\_chars) < 3:

return "Invalid Password"

# Password meets all the guidelines

return "Valid Password"

This function to check a password:

password = input("Enter password: ")

result = check\_password(password)

print(result)

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

1. Lambda functioJ

2. Filter functioJ

3. Zap functioJ

4. List ComprehensioI

\* Check if the string starts with a particular letterY

\* Check if the string is numericY

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

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

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

\* Check if a given number is evenY

\* Filter odd numbers from the given list.

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

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

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

1. the string starts with a particular letter:

string = "Hello, world!"

letter = "H"

# Using lambda function

starts\_with = lambda string, letter: string.startswith(letter)

print(starts\_with(string, letter))

# Using list comprehension

starts\_with = [string.startswith(letter) for string in [string]]

print(starts\_with)

1. if the string is numeric

string = "12345"

# Using lambda function

is\_numeric = lambda string: string.isnumeric()

print(is\_numeric(string))

# Using list comprehension

is\_numeric = [string.isnumeric() for string in [string]]

print(is\_numeric)

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

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

# Using lambda function

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

print(sorted\_fruits)

# Using list comprehension

sorted\_fruits = [fruit for fruit in sorted(fruits, key=lambda x: x[1])]

print(sorted\_fruits)

1. The squares of numbers from 1 to 10:

numbers = range(1, 11)

# Using lambda function

squares = list(map(lambda x: x\*\*2, numbers))

print(squares)

# Using list comprehension

squares = [x\*\*2 for x in numbers]

print(squares)

5.The cube root of numbers from 1 to 10:

import math

numbers = range(1, 11)

# Using lambda function

cube\_roots = list(map(lambda x: math.pow(x, 1/3), numbers))

print(cube\_roots)

# Using list comprehension

cube\_roots = [math.pow(x, 1/3) for x in numbers]

print(cube\_roots)

1. Check if a given number is even:

number = 8

# Using lambda function

is\_even = lambda number: number % 2 == 0

print(is\_even(number))

# Using list comprehension

is\_even = [number % 2 == 0 for number in [number]]

print(is\_even)

1. Filter odd numbers from the given list: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

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

# Using filter function and lambda function

odd\_numbers = list(filter(lambda x: x % 2 != 0, numbers))

print(odd\_numbers)

# Using list comprehension

odd\_numbers = [x for x in numbers if x % 2 != 0]

print(odd\_numbers)

1. Sort a list of integers into positive and negative integers lists: [1, 2, 3, 4, 5, 6, -1, -2, -3, -4, -5, 0]

numbers = [1, 2, 3, 4, 5, 6, -1, -2, -3, -4, -5, 0]

# Using lambda function

positive\_numbers = list(filter(lambda x: x > 0, numbers))

negative\_numbers = list(filter(lambda x: x < 0, numbers))

print(positive\_numbers)

print(negative\_numbers)

# Using