Here’s the solution to both Q1 and Q2.

**Q1: Password Validation Function**

import string

def check\_password(password):

# Check length

if len(password) != 10:

return "Invalid Password"

# Check for at least 2 uppercase letters

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

if uppercase\_count < 2:

return "Invalid Password"

# Check for at least 2 lowercase letters

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

if lowercase\_count < 2:

return "Invalid Password"

# Check for at least 1 digit

digit\_count = sum(1 for char in password if char.isdigit())

if digit\_count < 1:

return "Invalid Password"

# Check for at least 3 special characters

special\_chars = string.punctuation

special\_count = sum(1 for char in password if char in special\_chars)

if special\_count < 3:

return "Invalid Password"

return "Valid Password"

# Example usage

print(check\_password("Ab1!@#xyz")) # Output: Invalid Password

print(check\_password("ABcd12!@#$")) # Output: Valid Password

**Q2: Solve Questions with Lambda, Filter, Map, and List Comprehension**

1. **Check if the string starts with a particular letter (e.g., 'a'):**

starts\_with\_a = lambda s: s.startswith('a')

print(starts\_with\_a("apple")) # Output: True

print(starts\_with\_a("banana")) # Output: False

1. **Check if the string is numeric:**

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

print(is\_numeric("123")) # Output: True

print(is\_numeric("abc")) # Output: False

1. **Sort a list of tuples by quantity:**

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

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

print(sorted\_fruits) # Output: [('orange', 80), ('mango', 99), ('grapes', 1000)]

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

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

print(squares) # Output: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

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

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

print(cube\_roots) # Output: [1.0, 1.26, 1.44, 1.59, 1.71, 1.82, 1.91, 2.0, 2.08, 2.15]

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

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

print(is\_even(4)) # Output: True

print(is\_even(7)) # Output: False

1. **Filter odd numbers from a list:**

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

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

print(odd\_numbers) # Output: [1, 3, 5, 7, 9]

1. **Sort a list of integers into positive and negative integers:**

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

positive\_nums = sorted([x for x in nums if x > 0])

negative\_nums = sorted([x for x in nums if x < 0])

print("Positive:", positive\_nums) # Output: [1, 2, 3, 4, 5, 6]

print("Negative:", negative\_nums) # Output: [-5, -4, -3, -2, -1]