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”.

In [1]:
```
import re

def validate_password(password):

    # Define the password guidelines
    guidelines = [
        lambda p: len(p) >= 8,  # at least 8 characters long
        lambda p: any(c.isupper() for c in p),  # at least one uppercase letter
        lambda p: any(c.islower() for c in p),  # at least one lowercase letter
        lambda p: any(c.isdigit() for c in p),  # at least one digit
        lambda p: re.search(r"[^a-zA-Z0-9]", p) is not None  # at least one spec
    ]

    # Check if the password follows all guidelines
    if all(guideline(password) for guideline in guidelines):
        return "Valid Password"
    else:
        return "Invalid Password"

# Example usage:
password = input("Enter your password: ")
print(validate_password(password))
```

```
Valid Password
```

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

1. Lambda functioJ
2. Filter functioJ
3. Zap functioJ
4. List Comprehensiol

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]

1. Check if the string starts with a particular letter

In [2]:
```
def check_start_letter(string, letter):
    return string.startswith(letter)
```

```
# Using lambda function
check_start_letter_lambda = lambda string, letter: string.startswith(letter)

print(check_start_letter("Hello", "H"))  # True
print(check_start_letter_lambda("Hello", "H"))  # True
```

```
True
True
```

2. Check if the string is numeric

In [3]:

```
def check_numeric(string):
    return string.isnumeric()

# Using lambda function
check_numeric_lambda = lambda string: string.isnumeric()

print(check_numeric("12345"))  # True
print(check_numeric_lambda("12345"))  # True
```

```
True
True
```

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

In [4]:

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

# Using sorted function with lambda function as key
sorted_fruits = sorted(fruits, key=lambda x: x[1])

print(sorted_fruits)  # [("orange", 80), ("mango", 99), ("grapes", 1000)]
```

```
[('orange', 80), ('mango', 99), ('grapes', 1000)]
```

4. Find the squares of numbers from 1 to 10

In [5]:

```
# Using list comprehension
squares = [i**2 for i in range(1, 11)]

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

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

6. Check if a given number is even

In [6]:

```
def check_even(num):
    return num % 2 == 0

# Using lambda function
```

```
check_even_lambda = lambda num: num % 2 == 0

print(check_even(10))  # True
print(check_even_lambda(10))  # True
```

```
True
True
```

7. Filter odd numbers from the given list

In [8]:

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

# Using filter function with lambda function
odd_numbers = list(filter(lambda x: x % 2 != 0, numbers))

print(odd_numbers)  # [1, 3, 5, 7, 9]
```

```
[1, 3, 5, 7, 9]
```

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

In [9]:

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

# Using list comprehension
positive_numbers = [num for num in numbers if num > 0]
negative_numbers = [num for num in numbers if num < 0]

print(positive_numbers)  # [1, 2, 3, 4, 5, 6]
print(negative_numbers)  # [-1, -2, -3, -4, -5]
```

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

In [ ]: