Unit 3

Challenge 1 Q

Write a function called linear_search_product that takes the list o?f products and a target product name as input. The function should perform a linear search to find the target product in the list and return a list of indices of all occurrences of the product if found, or an empty list if the product is not found.

```
def linear_search_product(product_list,
target_product):
    indices = []

    for index, product in
enumerate(product_list):
        if product == target_product:
            indices.append(index)

    return indices

# Example usage:
products = ["Apple", "Banana", "Apple",
"Orange", "Mango", "Apple" ]
target = "Apple"

result = linear_search_product(products,
target)
print(result)
```

Output: [0, 2, 5]

Challenge 2 Q

Implement a function called sort_students that takes a list of student objects as input and sorts the list based on their CGPA (Cumulative Grade Point Average) in descending order. Each student object has the following attributes: name (string), roll_number (string), and cgpa (float). Test the function with different input lists of students.

```
class Student:
  def __init__(self, name, roll_number, cgpa):
    self.name = name
    self.roll_number = roll_number
    self.cgpa = cgpa

def sort_students(student_list):
  sorted_students = sorted(student_list,
key=lambda student: student.cgpa,
reverse=True)
  return sorted_students

# Example usage:
if __name__ == "__main__":
  students = [
    Student("Archana", "A101", 3.8),
    Student("Akshu", "B102", 3.5),
    Student("Nithi", "C103", 4.0),
    Student("Aruna", "D104", 3.9),
  ]

  sorted_students = sort_students(students)

  for student in sorted_students:
    print(f"Name: {student.name}, Roll
Number: {student.roll_number}, CGPA:
{student.cgpa}")
```

Output

Name: Nithi, Roll Number: C103, CGPA: 4.0

Name: Aruna, Roll Number: D104, CGPA: 3.9

Name: Archana, Roll Number: A101, CGPA: 3.8

Name: Akshu, Roll Number: B102, CGPA: 3.5