**Unit 3**

Unit - 3 Challenge

Challenge 1

Write a function called linear\_search\_product that takes the list of 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 = [index for index, product in enumerate(product\_list) if product == target\_product]

    return indices

# Example usage:

products = ["apple", "banana", "orange", "apple", "grape"]

target = "apple"

result = linear\_search\_product(products, target)

print(result)

Output:

[0, 3]

Challenge 2

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](http://self.name) = name

        self.roll\_number = roll\_number

        self.cgpa = cgpa

def sort\_students(student\_list):

    sorted\_students = sorted

Output:students = [

    Student("Alice", "001", 3.8),

    Student("Bob", "002", 3.5),

    Student("Charlie", "003", 4.0),

    # Add more students as needed

]

sorted\_students = sort\_students(students)

# Print sorted list

for student in sorted\_students:

    print(f"{[student.name](http://student.name)} ({student.roll\_number}): CGPA {student.cgpa}")

The output should be:Charlie (003): CGPA 4.0

Alice (001): CGPA 3.8

Bob (002): CGPA 3.

This indicates that the students are sorted based on their CGPA in descending order.