Assignment (End of Phase 1)

Instructions:

- Write Python programs to solve the following problems.
- Ensure your code is efficient and well-structured.
- Use appropriate comments to explain your logic.

Question 1: Inventory Management System

Create a Python program to manage a store's inventory. Your program should:

- Take inputs for different product names, their quantity, and price.
- Store them in appropriate variables and data structures.
- Use operators to calculate the total value of inventory.
- Allow the user to search for a product and display its details.
- If the searched product is unavailable, display an appropriate message.
- Use loops to allow multiple searches until the user decides to exit.
- Implement conditional statements to notify if stock is below a certain threshold (e.g., 5 units).

Example Output:

```
Enter number of products: 3
Enter product details (name, quantity, price):
Product 1: Apple, 10, 2.5
Product 2: Banana, 3, 1.2
Product 3: Orange, 5, 1.8
Total inventory value: $39.6
```

```
Search for a product: Banana
Product: Banana, Quantity: 3, Price: $1.2
Warning: Low stock!

Search for a product: Grapes
Product not found!
```

Question 2: Student Grade Calculator

Write a Python program that:

- Takes input for a student's name and their marks in five subjects.
- Uses variables to store this data.
- Uses operators to compute the total and average marks.
- Determines the grade using conditional statements based on the following criteria:
  - Average >= 90: Grade A
  - Average >= 80: Grade B
  - Average >= 70: Grade C
  - Average >= 60: Grade D
  - Below 60: Fail
- Displays the student's name, total marks, average, and grade.
- Use loops to allow multiple students' data entry until the user decides to stop.

Example Output:

```
Enter student name: Alice
Enter marks in 5 subjects: 88 92 79 85 90
Total Marks: 434
Average: 86.8
Grade: B
```

Question 3: Number Analysis Tool

Create a Python program that:

- Asks the user to input a list of numbers.
- Uses appropriate data types to store them.
- Uses loops and operators to calculate:
  - The sum of all numbers.
  - The average of numbers.
  - The maximum and minimum number.
- Uses conditional statements to check whether the sum of numbers is even or odd.

Example Output:

```
Enter numbers separated by space: 10 15 20 25 30
Sum: 100
Average: 20.0
Maximum: 30
Minimum: 10
Sum is Even.
```

Question 4: Password Strength Checker

Write a Python program that:

- Asks the user to enter a password.
- Uses conditional statements to check the strength of the password based on these conditions:
  - At least 8 characters long.
  - Contains both uppercase and lowercase letters.
  - Contains at least one digit.
  - Contains at least one special character (!, @, #, $, etc.).

- Uses loops to allow multiple attempts until a strong password is entered.

Example Output:

```
Enter password: weakpass
Weak password! Try again.
Enter password: Strong@123
Password is strong!
```

End of Test

Answer 1: Inventory Management System

```
# Inventory Management System

# Step 1: Get the product details from user
inventory = {}  # empty list

num_of_products = int(input("Enter number of products: "))

print("Enter product details (name, quantity, price): ")

for i in range(num_of_products):
    name = input(f"Enter Product {i+1} Name: ").strip()
    quantity = int(input(f"Enter quantity of {name}: "))
    price = float(input(f"Enter price of {name}: "))
    inventory[name] = {"quantity": quantity, "price": price}

# Step 2: Calculate total inventory value
total_value = 0
for item in inventory.values():  # values() function
    total_value += item["quantity"] * item["price"] # Add each product's total cost
print(f"\nTotal inventory value: ${total_value: .2f}")



# Step 3: Search functionality with loop
```

```
while True:
    search_text = input("\nSearch for a product (or type 'exit' to stop): ").strip()

    if search_text.lower() == "exit":
        break

    if search_text in inventory:
        item = inventory[search_text]
        print(f"Product: {search_text}, Quantity: {item['quantity']}, Price: ${item['pri
        if item["quantity"] < 5:
            print("Warning: Low Stock!")
    else:
        print("Product not found!")
```