**Problem:**

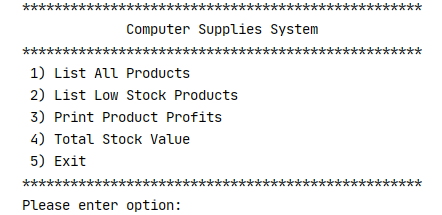
**Note: There is no need for any returns in any of the functions.**

Create a system for a Computer Supplies shop, which manages stock.

All functions defined should be in the **same** file as the main body of the program.

The program should have a menu to select options, where after each option except exit, the program redisplays the menu.

The menu should look similar to:



• The **menu printing** should be written as a **function.**

• Copy the following 2D list into your program, **outside** of any functions.

product\_list = [[**"CS567"**, **"Wireless Printer"**, 49.99, 20.00, 8, 10],  
 [**"CS100"**, **"Document Scanner"**, 109.99, 60.00, 11, 5],  
 [**"CS777"**, **"Ink Cartridge"**, 29.99, 15.00, 12, 25],  
 [**"CS800"**, **"Full HD Webcam"**, 64.00, 30.00, 12, 10],  
 [**"CS990"**, **"Optical Mouse"**, 5.99, 2.00, 10, 5]]

The above 2D list stores products’ details as follows:

Column 1: Product ID

Column 2: Name

Column 3: Selling price

Column 4: Cost price

Column 5: Quantity of the product on hand (i.e. in the shop)

Column 6: Re-order level (a quantity which indicates that the product should be re-ordered)

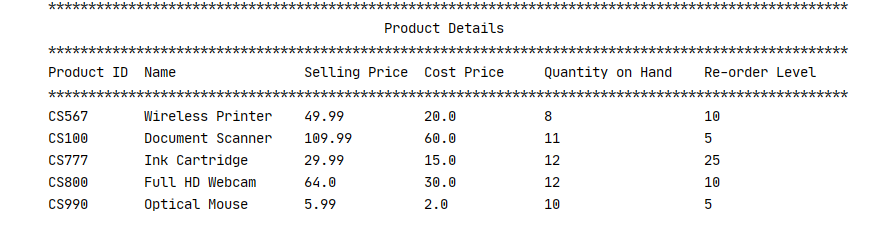
**Option 1: List All Products**

This option displays all the products and their details in a formatted manner.

This logic should be coded in a function, **display\_products**().

When this option is selected from the menu, the **display\_products()** function should be called and the **product\_list** should be passed to it as an argument.

**Sample Output:**

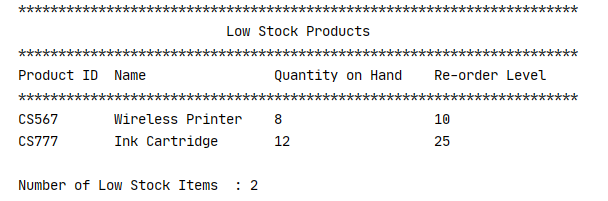


**Option 2: List Low Stock Products**

When this option is selected from the menu the products which are at or below their reorder level should be displayed in a formatted manner, you need only print the product ID, name, quantity on hand and re-order level. This logic should be coded in a function, **list\_low\_stock().** The function should also calculate and print the number of low stock products as per sample output **below**.

When this option is selected from the menu, the **list\_low\_stock ()** function should be called and the **product\_list** should be passed to it as an argument.

**Sample Output:**

****

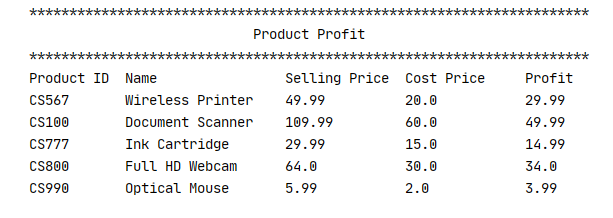
**Option 3. Print Product Profits**

When the user selects this option, the program should call a function **calculate\_profit(),** and pass the 2D **product\_list** into the **calculate\_profit()** function as an argument.

The **calculate\_ profit ()** function calculates and prints the profit made on each product. It should print the product ID, name, selling price, cost price and profit.

There is no need to store the profit in the list, simply print it.

**Sample Output:**



**Option 4. Total Stock Value**

When the user selects this option, the program should call a function **calculate\_stock\_value(),** and pass the 2D **product\_list** into the **calculate\_stock\_value ()** function as an argument.

This function calculates and prints the total stock value of the stock that the company has in the shop. Use the **selling price** in this calculation.

**Sample Output:** 

Output should be presented neatly but does not have to be exactly as above.

**Deliverables:**

1. A Python program which solves the above problem.