Dear students,

Here's another activity to ensure that you remember and don't forget the topics we've learnt so far. Similar to the previous activity, I won't give you detailed instructions on how to do this, I'll only give you the problem statement and some broad hints so that you can figure out how best to approach solving the problem.

**Problem Statement**

**Part 1**

Your task is to implement a small e-commerce feature - **updating the cart price** when adding a new item to a shopping cart. The cart may or may not already have other items, so, **the initial price of the cart can be any number greater than or equal to zero** (but assume that it's **less than 500 for reasons you'll see below**). You can hard-code the initial value of the cart.

When you add a new item to the cart (the price of which is an input), the total price of the cart must be updated to reflect the addition of the item. There is a catch - if the **total price of the cart exceeds** (i.e., is greater than) **Rs. 500**, then you need to apply a flat **7.5% discount** to the entire cart. Note that this discount must be applied **AFTER** adding the price of the new item to the total and not before.

**Part 2**

Now, just make a slight adjustment to what you've implemented above. Instead of entering the total price of the new item, enter the **unit price** and the **quantity** of the item. **Calculate the total price of the item** based on these two variables. Then, use this calculated total price of the item to **update the price of the cart**. As before, the **7.5% discount** applies if the total cart price after adding the new item exceeds Rs. 500.

**Part 3**

Add a **loop** to keep getting the prices of **multiple items** until the user says they don't have any more items to add (you need a **separate variable** to keep track of the **user's choice**). Each time a **new item is added, the cart price must be updated**, and **finally**, after all the items have been added, the **7.5% discount** must be calculated and applied if the total price is greater than Rs. 500.

**Write a C++ program to implement the above feature using suitable variables, functions, and control structures.**

**Ensure the following in your code**

- **Code readability** (correct indentation, meaningful variable and function names, spacing between code blocks, etc)
- Good **user experience** (clear prompts for inputs)
- **Appropriate data types** for your variables (prices usually have decimal points)

**Hints**

- Create separate functions to calculate the price of an item based on the unit price and quantity, and to calculate the cart total based on the previous cart total and the new item price, and perhaps even to apply the discount
- Use the main function to initialise the value of the cart and read inputs and display outputs, but delegate the calculations to separate functions as mentioned above
- Remember, **percentages** can be represented as **decimals**
- **Quantity** may be a **whole number** for items sold in discrete quantities such as smartphones *(you can't buy half a smartphone)*, but it can also be **fractional numbers** for other types of items *(e.g., you can buy 2.5kg of rice!)*
- The way the problem statement is written, you don't really need to know the name of the item you're adding to the cart, just the price *(when we get to object-oriented programming concepts, we'll see how we can deal with that)*, so, only focus on what you need to do for this activity

If you have any questions, please don't hesitate to reach out via WhatsApp, I'm happy to help and provide feedback.

Happy coding!

Kind regards,
Walter