CSE108 – Computer Programming Lab

Lab 12

Dynamic Arrays
24/05/2025

You will implement a simple shopping cart using dynamic arrays in C. The cart contains a list of items. Each item has a name, a price, and a quantity. The cart itself also stores the number of **distinct** items currently in the cart (called *count*) and the total capacity of the cart — i.e., how many items it can hold before needing more memory.
As users add items, the system should automatically resize the internal array to fit the new items. At the end of the program, all allocated memory must be freed.

Note: The cart's "count" refers to the number of distinct items in the cart, not the total quantity of all items.

Part 1. (10 pts) You will implement a function that initializes the shopping cart. The user enters an initial capacity (e.g., 2), which determines how many items the cart can initially hold.

Part 2. (30 pts) You will implement a function to add a new item to the cart. The user will be prompted to enter:

- The name of the item (e.g., "Milk")
- The price of the item (e.g., 4.50)
- The quantity to add (e.g., 2)

If the number of items in the cart reaches its current capacity, you will **double the capacity** of the array before adding the new item.

Part 3. (15 pts) You will implement a function to search for an item by name. If found, return its index and print the item details. If the item is not found, display an appropriate message. (e.g., "Item not found").

Part 4. (20 pts) You will implement a function to remove an item by name. If found, shift all items after it one position to the left and reduce the count. If not found, print an error message. If the number of items falls below half the capacity, shrink the array and update the capacity.

Part 5. (15 pts) You will implement a function that prints the cart contents in a table format. For each item, show name, price, quantity, and subtotal (price * quantity). At the bottom, print the total value of the cart.

Part 6. (10 pts) You will implement a function that frees the memory used by the cart at the end of the program using the free function.

RULES:

1. Using fixed-sized arrays is not allowed.
2. Global variables are not allowed.
3. Display the cart in a clean tabular format with two decimal places for monetary values.

```
Enter initial cart capacity: 2

1. Add Item
2. Remove Item
3. Search Item
4. Print Cart
5. Exit

Choice: 1
Enter item name: Milk
Enter price: 5.00
Enter quantity: 2
Item added.

Choice: 1
Enter item name: Bread
Enter price: 3.00
Enter quantity: 1
Item added.

Choice: 1
Enter item name: Eggs
Enter price: 2.50
Enter quantity: 6
Array resized to capacity: 4
Item added.

Choice: 4
------------------------------------------------------------
Name          Price         Quantity            Subtotal
------------------------------------------------------------
Milk          5.00          2                   10.00
Bread         3.00          1                   3.00
Eggs          2.50          6                   15.00
------------------------------------------------------------
Total cart value: 28.00

Choice: 2
Enter item name to remove: Bread
Item removed.

Choice: 4
------------------------------------------------------------
Name          Price         Quantity            Subtotal
------------------------------------------------------------
Milk          5.00          2                   10.00
Eggs          2.50          6                   15.00
------------------------------------------------------------
Total cart value: 25.00

Choice: 5
Freeing memory... Goodbye!
```

→ add
benim itemim sığıyor mu?
count > capacity
→ double capacity
else
just add