

University of Bouira  
Department of Computer Science

2024-2025

## Inventory Management System with Recursive Search and Sorting

### Instructions:

- This is an individual test. No collaboration is allowed.
- You have **45 minutes** to complete the test.
- Use a modular design approach, breaking down your solution into distinct functions where possible.
- Clearly comment your code to explain your logic.
- Partial credit will be awarded for correct code components, even if the full program is not completed.

## Problem Statement

You are tasked with creating a **Inventory Management System** in C. The system should allow users to manage inventory items using structures, pointers, dynamic memory allocation, and recursion. The following requirements must be met:

### 1. Define a Structure for Inventory Items (0.5 points)

Define a structure `Item` with the following fields:

- `itemCode`: a character array of maximum length 10 to store the item code.
- `itemName`: a character array of maximum length 50 to store the item name.
- `quantity`: an integer to store the quantity of the item.
- `price`: a float to store the price of the item.

## 2. Dynamic Array for Inventory (1 point)

- Prompt the user to enter the number of items in the inventory.
- Dynamically allocate memory for an array of `Item` structures of this size.
- Create a function that uses pointers to populate this array with item details (item code, item name, quantity, and price).

## 3. Recursive Search Function (1 point)

Write a recursive function `searchItem` that:

- Takes an item code as input.
- Recursively searches for the item with the given code in the inventory array.
- The function should return a pointer to the matching item or `NULL` if no match is found.

## 4. Recursive Sorting Function (Quick Sort) (1.5 points)

Implement a recursive Quick Sort function to sort inventory items by `price` in ascending order. If two items have the same price, they should be sorted alphabetically by `itemName`.

- The function should take pointers to the start and end of the array and sort the items in place.

## 5. Display Function (1 point)

Write a function `displayInventory` to display the sorted inventory in a readable format. The display should include:

- Item Code
- Item Name
- Quantity
- Price

## 6. Main Menu (1 point)

Implement a menu with the following options:

1. Add item details (expand the array if needed).
2. Search for an item by code (using the recursive search function).
3. Sort and display all items by price (using the recursive sorting function).
4. Exit.