University of Bouira
Department of Computer Science

2024-2025

Contact Management System Using Structures and Recursion

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 simple **Contact Management System** in C. The system should allow users to manage contact information using structures, pointers, dynamic memory allocation, and recursive functions. The following requirements must be met:

1. Define a Structure for Contacts (0.5 points)

Define a structure `Contact` with the following fields:

- `firstName`: a character array of maximum length 50.
- `lastName`: a character array of maximum length 50.
- `phoneNumber`: a character array of maximum length 15.
- `email`: a character array of maximum length 50.

2. Dynamic Array for Contacts (1 points)

- Prompt the user to enter the initial number of contacts they want to add.
- Dynamically allocate memory for an array of `Contact` structures of this size.
- Create a function that uses pointers to populate this array with contact details from user input.

3. Recursive Search Function (1 points)

Write a recursive function `searchByName` that:

- Takes a partial name (either first or last name) as input.
- Recursively searches for matches in the contact array.
- Prints any matching contacts or a message if no matches are found.

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

Implement a recursive Quick Sort function to sort contacts by `lastName`. If two contacts have the same last name, they should be sorted by `firstName`.

5. Delete Contact by Index (1 points)

Write a function `deleteContact` that:

- Takes an index as input and removes the contact at that position.
- Shifts all subsequent contacts up by one position to fill the gap.
- Reallocates memory to resize the array, if necessary.

6. Menu for Operations (1 points)

Implement a menu with the following options:

1. Add new contact details (expand the array if needed).
2. Search contacts by name (using the recursive search function).
3. Sort and display all contacts by last name.
4. Delete a contact by index.
5. Exit.