NAMIBIA UNIVERSITY OF SCIENCE AND TECHNOLOGY

DATA STRUCTURE AND ALGORITHM 1: Group Assignment

Group 75 members

Name and Surname	Student Number
Muingona Katjizeu (group leader)	224061763
Rowan Van Wyk	224002244
Kwame Tambrescu	224058509
Jeysen Nyandoro	224002058
Mickylla Visagie	223105090
Johannes Eelu	223127531

SECTION A

1. THE PHONEBOOK MODULES AND FUNCTIONS

1. Contact Management

This is the core module it handles the phone book's most basic operations that users expect and wish to perform. The module's purpose is to allow users to manage their contacts in the most efficient way, most straightforward, and most common functionalities as a guideline.

FUNCTIONS

-Insert Contact: Adds a new contact to the phonebook.

Input: Contact Name, Phone Number

Process: Add contact to the array/list

Output: Confirmation of insertion

-Search Contact: Finds a contact by name or number.

Input: Contact Name

Process: Search for contact in the array/list

Output: Contact details or "Not found"

-Delete Contact: Removes a contact from the phonebook.

Input: Contact Name

Process: Search and remove contact from the array/list

Output: Confirmation of deletion

-Update Contact: Modifies the details of an existing contact.

Input: Contact Name, New Phone Number

Process: Search for contact and update number

Output: Confirmation of update

-Display All Contacts: Lists all contacts in the phonebook.

Process: Iterate through the array/list

Output: List of all contacts

2. Data Persistence Module

For any application that deals with user identity, long-term data storage is a must. The trust that users place in you when they allow you to manage their identities is critical, and anything that compromises it could have outcomes. This module is responsible for the storage and retrieval of the application's data model, implementing the necessary functionality for both processes.

Functions

-Save Contacts to File

-Load Contacts from File

-Backup Contacts

3.Performance evaluation module

In this module, we understand how the application performs, especially when the number of contacts expands. We look at efficiency in terms of how well or poorly the application can scale. Scaling is essential for telecommunications applications

since the number of users can easily reach colossal sizes. We analyse how well the search function works, using the same tests and measurements as before. We focus on time measurements to see if the search algorithm can find contacts in a reasonable time, even when the set of contacts is large.

Functions

-Record search time

-Analyse search

-Calculate average search times

4.Sorting (Optional)

Sorting enhances the usability of the phonebook, making it easier for users to find contacts quickly, especially as the list grows. While not strictly necessary, it improves the application's performance during search operations when combined with efficient searching algorithms. It is responsible for sorting the contacts to improve search efficiency. It can implement various sorting algorithms to organize the contacts based on names or other attributes.

FUNCTIONS

-Process: Sort the array/list of contacts

-Output: Sorted list of contacts

2. THE PSEUDOCODE

START

i = 0

WHILE (i < 8)

```
    PROMPT USER FOR NAME, PHONE NUMBER AND EMAIL
    GET NAME, PHONE NUMBER AND EMAIL
    IF (i == 1) THEN
        DISPLAY "CONTACT ADDED"
    ELSE IF (i == 2) THEN
        DISPLAY "CONTACT DETAILS"
    ELSE IF (i == 3) THEN
        DISPLAY "ALL CONTACTS"
    ELSE IF (i == 4) THEN
        DISPLAY "CONTACT DELETED"
    ELSE IF (i == 5) THEN
        DISPLAY "CONTACT UPDATE"
    ELSE IF (i == 6) THEN
        DISPLAY "PHONEBOOK SORTED BY NAME"
    ELSE IF (i == 7) THEN
        DISPLAY "EXITING"
    ELSE
        DISPLAY "INVALID NUMBER"
    ENDIF
    i = i + 1
ENDWHILE
END
```