DATA STRUCTURES (MCA 101) PROJECT

SARATH PRASAD T S

MCA

TOPIC: PHONE BOOK USING DATA STRUCTURES

PROJECT DETAILS

1. PROJECT TITLE:PHONE BOOK

2.Project Aim:

The aim of this project is to create a phonebook management system in C, allowing users to store, retrieve, update, and delete contact information. The program provides a user-friendly interface for managing contacts efficiently.

3.Functionality:

- **Insertion:** Allows users to add new contacts with names and phone numbers.
- **Display:** Displays all contacts in a tabular format.
- **Search:** Enables users to search for a contact by name**.**
- **Deletion:** Allows users to delete a contact by name.
- **Update:** Provides options to update either the name or phone number of an existing contact.

4.Data Structure Used:

The data structure used for the implementation is a singly linked list. Each node in the linked list contains two strings: one for the contact name and another for the phone number**.**

5.Reason for Choosing Singly Linked List:

A singly linked list is chosen because it allows dynamic memory allocation, meaning the program can handle an indefinite number of contacts. It also provides efficient insertion and deletion operations, crucial for a phonebook management system where contacts are frequently added or removed.

6.Novelty/Creativity in Your Project:

1. User-Friendly Interface: project provides a simple and intuitive menu-driven interface for users, making it easy to interact with the phonebook system.

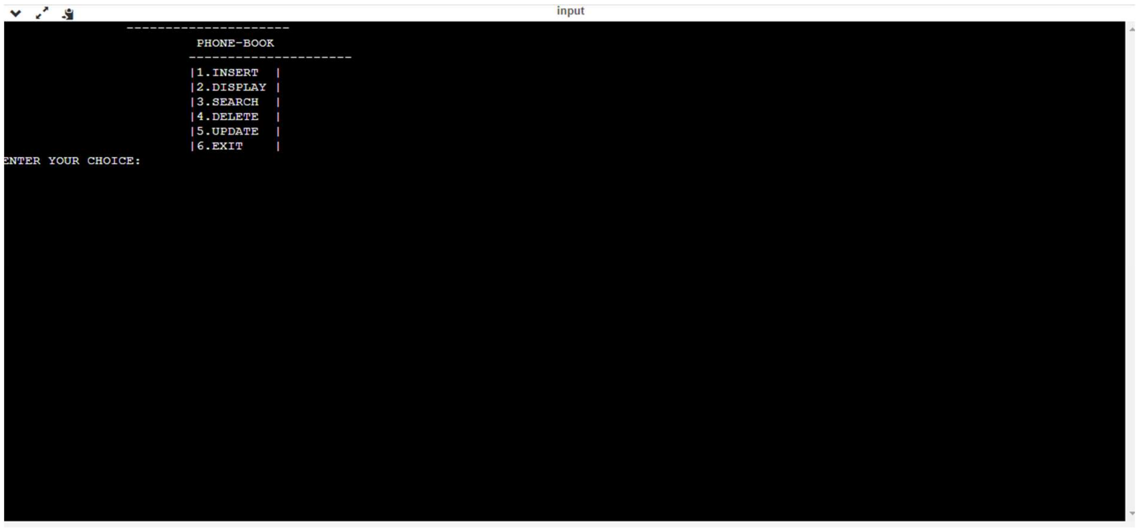
2. Dynamic Memory Allocation: By using a singly linked list, your implementation allows for dynamic memory allocation, enabling the program to handle a varying number of contacts without a predefined size limit.

3. Error Handling: program checks for duplicate names when adding new contacts and handles cases where the user tries to update a name to an existing one, providing a rudimentary form of error prevention.

4. Efficient Searching:it efficiently finds and displays the contact information when a match is found, contributing to a smoother user experience**.**

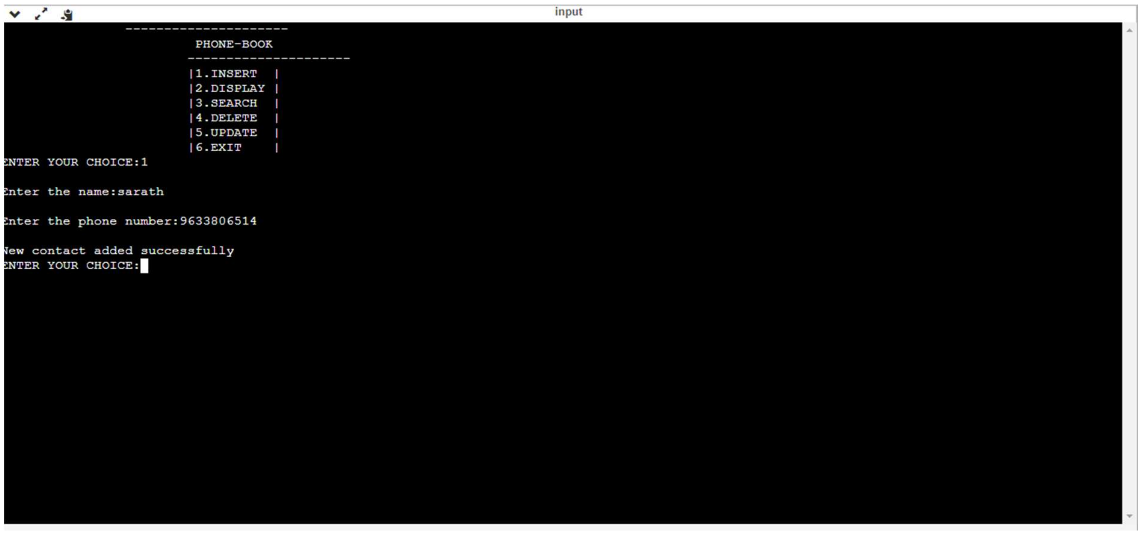
5. Modularity: This program is modularized, with separate functions for different operations, enhancing code readability and maintainability.

WORKING OF PROGRAM



```
                --------------------
                        PHONE-BOOK
                       --------------------
                       |1.INSERT  |
                       |2.DISPLAY |
                       |3.SEARCH  |
                       |4.DELETE  |
                       |5.UPDATE  |
                       |6.EXIT    |
ENTER YOUR CHOICE:
```

1. Insertion (Adding a New Contact):



```
                --------------------
                         PHONE-BOOK
                        --------------------
                        |1.INSERT  |
                        |2.DISPLAY |
                        |3.SEARCH  |
                        |4.DELETE  |
                        |5.UPDATE  |
                        |6.EXIT    |
NTER YOUR CHOICE:1

nter the name:sarath

nter the phone number:9633806514

ew contact added successfully
NTER YOUR CHOICE:
```

1. User Input:

- The program prompts the user to enter the name and phone number of the contact they want to add.

- The user provides the name and phone number as input.

2. Insert Function:

- The program calls the `insert()` function, passing the entered name and phone number as parameters.

- In the `insert()` function:

  - It checks if the phonebook is empty (head is `NULL`).

  - If empty, it creates the first node for the contact and assigns the entered name and phone number.

  - If not empty, it traverses the existing nodes:

    - If the name already exists in the phonebook, it displays a message indicating that the name is already taken.

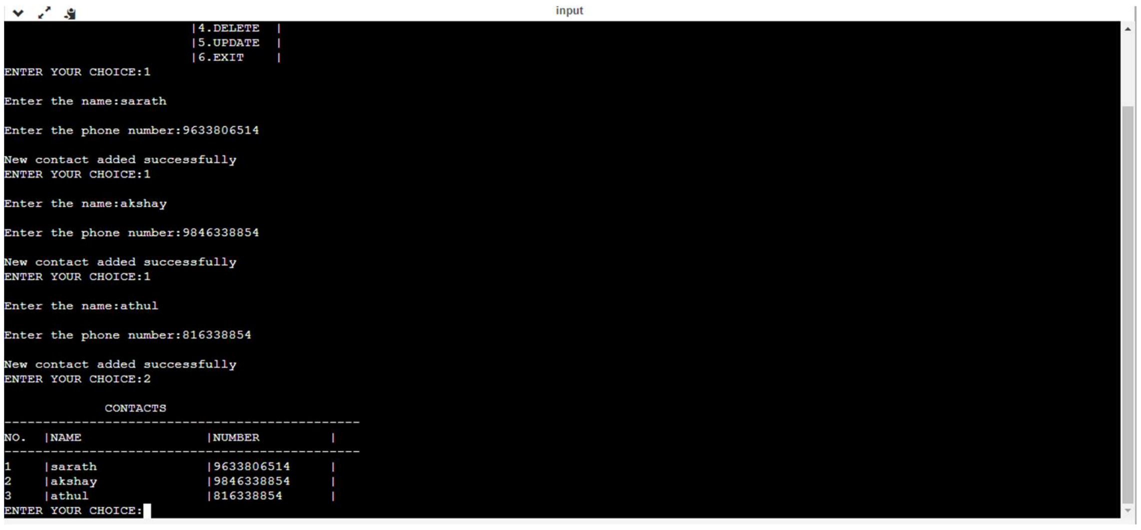
    - If the name is unique, a new node is created, and the name and phone number are assigned.

    - The new node is added at the end of the linked list.

3. **Output (Insertion Confirmation):**

The program displays a message confirming that the contact has been added successfully.

Displaying (Viewing All Contacts):



```
                        |4.DELETE  |
                        |5.UPDATE  |
                        |6.EXIT    |
ENTER YOUR CHOICE:1

Enter the name:sarath

Enter the phone number:9633806514

New contact added successfully
ENTER YOUR CHOICE:1

Enter the name:akshay

Enter the phone number:9846338854

New contact added successfully
ENTER YOUR CHOICE:1

Enter the name:athul

Enter the phone number:816338854

New contact added successfully
ENTER YOUR CHOICE:2

             CONTACTS
----------------------------------------------
NO.  |NAME               |NUMBER          |
----------------------------------------------
1    |sarath             |9633806514      |
2    |akshay             |9846338854      |
3    |athul              |816338854       |
ENTER YOUR CHOICE:
```

1. User Request:

- The user chooses the "Display" option from the menu.

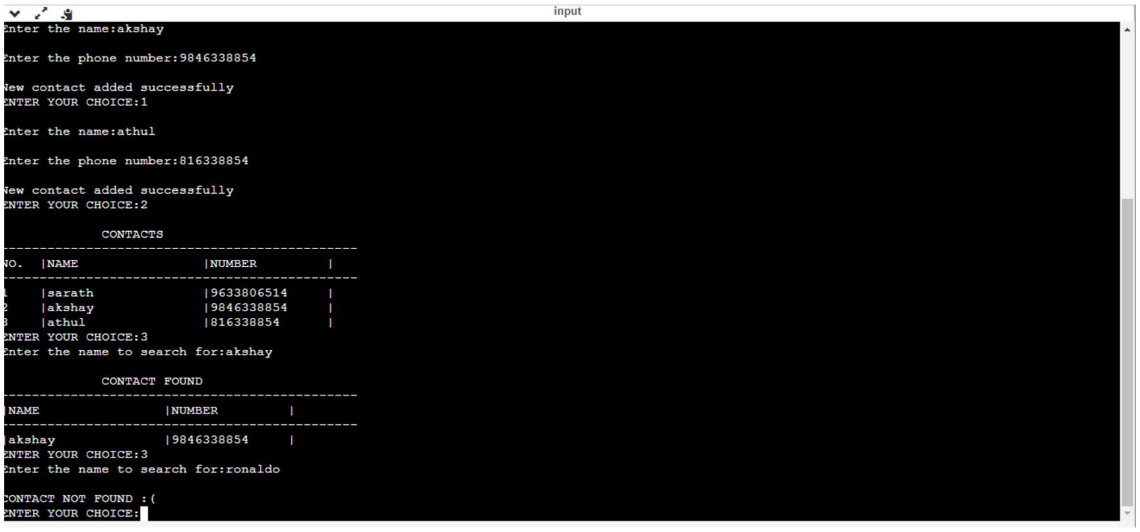
2. Display Function:

- The program calls the `display()` function.
- In the `display()` function:
  - It checks if the phonebook is empty (head is `NULL`).
  - If empty, it displays a message indicating that no contacts are found.
  - If not empty, it iterates through the linked list and displays the contacts in a tabular format, including contact number, name, and phone number.

3. Output (Displaying Contacts):

- The program outputs a table with contact details, including the contact number, name, and phone number, for all contacts present in the phonebook.

Searching for a Contact:



```
nter the name:akshay

nter the phone number:9846338854

ew contact added successfully
NTER YOUR CHOICE:1

nter the name:athul

nter the phone number:816338854

ew contact added successfully
NTER YOUR CHOICE:2

            CONTACTS
----------------------------------------------
O.   |NAME                |NUMBER          |
----------------------------------------------
     |sarath              |9633806514      |
     |akshay              |9846338854      |
     |athul               |816338854       |
NTER YOUR CHOICE:3
nter the name to search for:akshay

            CONTACT FOUND
----------------------------------------------
|NAME                |NUMBER          |
----------------------------------------------
|akshay              |9846338854      |
NTER YOUR CHOICE:3
nter the name to search for:ronaldo

ONTACT NOT FOUND :(
NTER YOUR CHOICE:
```

1. User Input:

- The program prompts the user to enter the name of the contact they want to search for.
- The user provides the name as input.

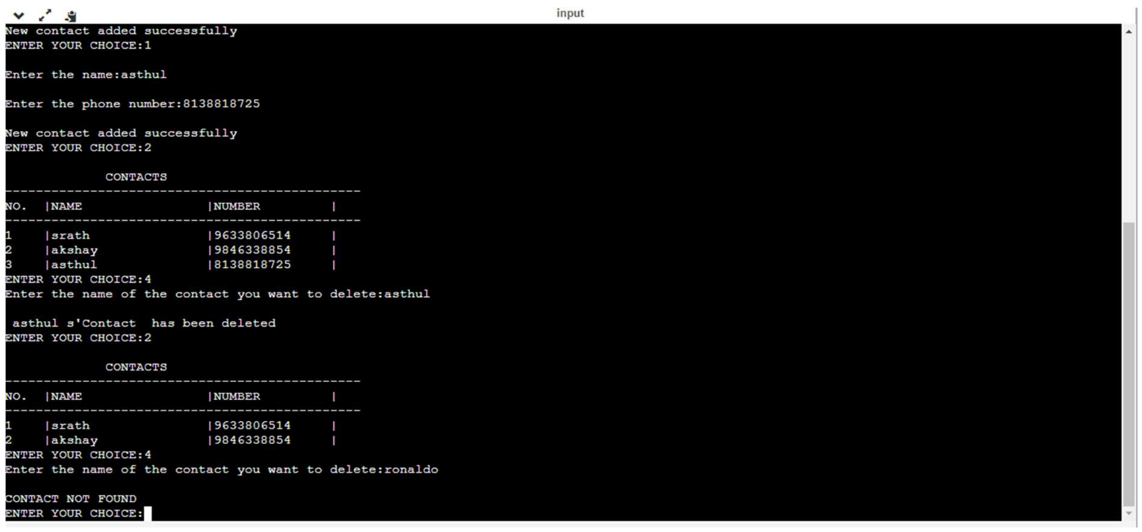
2. Search Function:

- The program calls the `search()` function, passing the entered name as a parameter.
- In the `search()` function:
  - It traverses the linked list, comparing the entered name with the names of contacts in the phonebook.
  - If a contact with a matching name is found, the program displays the contact's details (name and phone number).
  - If no matching contact is found, the program displays a message indicating that the contact was not found.

3. Output (Search Results):

- If a matching contact is found:
  - The program outputs the contact's name and phone number, confirming that the contact was found.
- If no matching contact is found:
  - The program outputs a message indicating that the contact was not found in the phonebook.

Deleting a Contact:



```
New contact added successfully
ENTER YOUR CHOICE:1

Enter the name:asthul

Enter the phone number:8138818725

New contact added successfully
ENTER YOUR CHOICE:2

             CONTACTS
----------------------------------------------
NO.  |NAME                |NUMBER         |
----------------------------------------------
1    |srath               |9633806514     |
2    |akshay              |9846338854     |
3    |asthul              |8138818725     |
ENTER YOUR CHOICE:4
Enter the name of the contact you want to delete:asthul

 asthul s'Contact  has been deleted
ENTER YOUR CHOICE:2

             CONTACTS
----------------------------------------------
NO.  |NAME                |NUMBER         |
----------------------------------------------
1    |srath               |9633806514     |
2    |akshay              |9846338854     |
ENTER YOUR CHOICE:4
Enter the name of the contact you want to delete:ronaldo

CONTACT NOT FOUND
ENTER YOUR CHOICE:
```

1. User Input:

- The program prompts the user to enter the name of the contact they want to delete.
- The user provides the name as input.

2. Delete Function:

- The program calls the `delete()` function, passing the entered name as a parameter.
- In the `delete()` function:
  - It traverses the linked list, comparing the entered name with the names of contacts in the phonebook.
  - If a contact with a matching name is found:
    - If it's the first node, it updates the `head` pointer to skip the first node (effectively deleting it).
    - If it's not the first node, it updates the `next` pointers to skip the node with the matching name (effectively deleting it).
  - If no matching contact is found, the program displays a message indicating that the contact was not found in the phonebook.

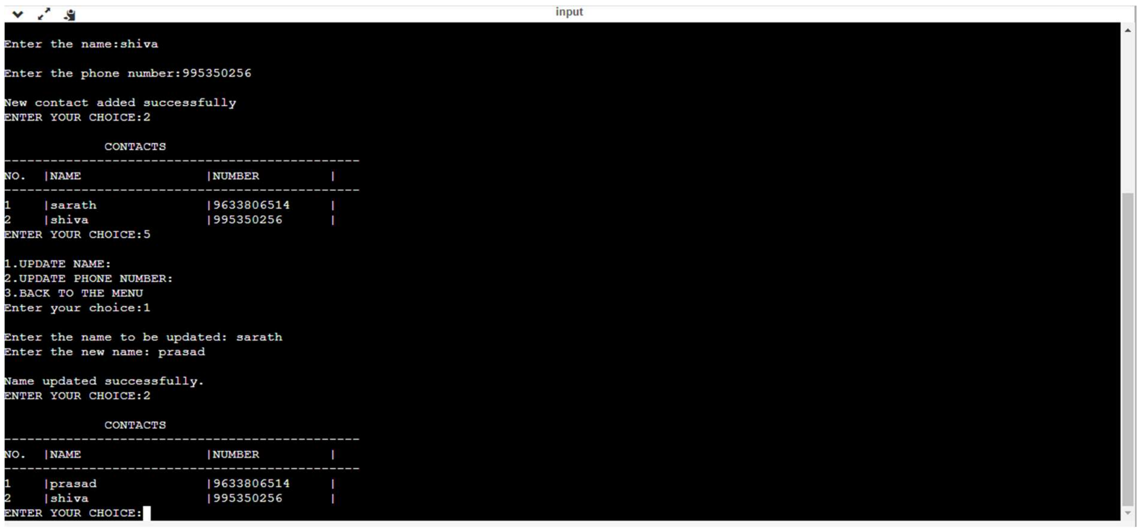
3. Output (Deletion Confirmation or Error Message):

- If a matching contact is found and deleted:
  - The program outputs a confirmation message indicating that the contact was successfully deleted.

- If no matching contact is found:

  - The program outputs a message indicating that the contact was not found in the phonebook and thus could not be deleted.

Updating a Contact:



```
Enter the name:shiva

Enter the phone number:995350256

New contact added successfully
ENTER YOUR CHOICE:2

             CONTACTS
----------------------------------------------
NO.  |NAME                |NUMBER         |
----------------------------------------------
1    |sarath              |9633806514     |
2    |shiva               |995350256      |
ENTER YOUR CHOICE:5

1.UPDATE NAME:
2.UPDATE PHONE NUMBER:
3.BACK TO THE MENU
Enter your choice:1

Enter the name to be updated: sarath
Enter the new name: prasad

Name updated successfully.
ENTER YOUR CHOICE:2

             CONTACTS
----------------------------------------------
NO.  |NAME                |NUMBER         |
----------------------------------------------
1    |prasad              |9633806514     |
2    |shiva               |995350256      |
ENTER YOUR CHOICE:
```

1. User Input:

- The program prompts the user to enter the name of the contact they want to update.
- The user provides the name as input.

2. Update Function:

- The program calls the `update()` function, initiating the update process.
- In the `update()` function:
  - It first asks the user whether they want to update the contact's name or phone number.
  - If the user chooses to update the name:
    - The program prompts the user to enter the new name.
    - It checks if the new name is already taken. If yes, it displays an error message.
    - If the new name is unique, it updates the name of the contact.
  - If the user chooses to update the phone number:
    - The program prompts the user to enter the new phone number.
    - It updates the phone number of the contact.
  - If the user chooses to go back to the menu, the function exits without performing any updates.

3. Output (Update Confirmation or Error Message):

- If the update is successful, the program outputs a confirmation message indicating that the contact's information was updated.

- If the new name is not unique (for name updates), the program displays an error message indicating that the name is already taken.

- If the contact to be updated is not found, the program displays a message indicating that the contact was not found in the phonebook.