A Project Report
On

PHONE DIRECTORY

Submitted by

KHYATI SINGH (RA2211026030012)
HARSHIT SINGH (RA2211026030028)
DIYA MISHRA (RA2211026030041)
NAMAN SWAMY (RA2211026030053)
BHAVIKA SINGH (RA2211026030058)

Under the Supervision of
MS. NIDHI PANDEY
(Assistant Professor, Department of Computer Science and Engineering)

SRM
INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)
DELHI-NCR CAMPUS, GHAZIABAD (U.P)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
DELHI-NCR CAMPUS, MODINAGAR
SIKRI KALAN, DELHI MEERUT ROAD,
DIST. – GHAZIABAD - 201204

Odd Semester- 2023-24

SRM INSTITUTE OF SCIENCE TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "PHONE DIRECTORY" is the bonafide work of "KHYATI SINGH (RA2211026030012), HARSHIT SINGH (RA2211026030028), DIYA MISHRA (RA2211026030041), NAMAN SWAMY (RA2211026030053), BHAVIKA SINGH (RA2211026030058)", who carried out the project work under my supervision. Certified further, that to the best of my knowledge, the work reported here does not form any other project report or dissertation.

SIGNATURE SIGNATURE

Course-In-Charge (HEAD OF DEPARTMENT)
MS. NIDHI PANDEY
(Assistant Professor)

Dept. of Computer Science and Engineering

INDEX

ABSTRACT

Designed to make contact management simple for people, the Phone Directory Application is a user-friendly and effective application. With features including generating, adding, displaying, removing, searching, and sorting contacts inside a linked list, the program helps users manage a list of contacts. A summary of the application's code, hardware requirements, goals, pros and cons, approach, findings, and references are given in the report. Because of its simple and intuitive user interface, this program is usable by users of all skill levels, including those who are novices. The Phone Directory Application is a useful tool for anyone trying to keep their contact list organized, be they a student studying computer technology or just an individual.

INTRODUCTION

An orderly and user-friendly contact management system is more important than ever in the current digital era. This need is met by the Phone Directory Application, which offers an easy-to-use and effective platform for managing your contacts. This application is meant to make the process of managing your contacts easier, regardless of whether you're a computer science student starting your journey to learn programming languages or just someone looking for a better way to organize your contacts.

Since the invention of smartphones, the majority of us always have a long list of contacts on hand. Finding, editing, and arranging these contacts, however, can frequently be a laborious and time-consuming process.

This procedure is intended to be streamlined by the Phone Directory Application so that you may easily manage your contacts. Beginners can easily navigate its user-friendly interface, which removes the learning curve that comes with using many contact management programs. This program stores contact details, such as first and last names and phone numbers, in a doubly linked list data structure.

OBJECTIVE

Phone Directory: We want to be able to add, update, and delete phone numbers and their associated contact information.

- **Efficiency:** It's important that our system is superfast and doesn't waste any memory. We'll be using clever techniques to make this happen.
- **Search and Retrieval:** You'll be able to quickly find a contact by searching for their phone number.
- **User Interface:** We're designing an easy-to-use interface that will allow you to do all the things you'd expect - like adding new contacts or editing existing ones.
- **Data Persistence:** We want to make sure your contacts are safe and sound, even when you close the program and reopen it later.
- **Data Validation:** We'll help you enter phone numbers and contact information correctly, so there are no mistakes.
- **Sorting and Display:** You can sort your contacts and display them in a way that makes the most sense to you.
- **Security:** Your data will be secure, and only authorized users will be able to access it.
- **Error Handling:** If something goes wrong, we're going to make sure the program doesn't crash - it'll handle problems gracefully.
- **Scalability:** Our system will grow with you. You can add as many contacts as you need without slowing it down.
- **Documentation:** We'll provide clear instructions on how to use the phone directory, so you won't get lost.
- **Testing and Quality Assurance:** We're going to test this thoroughly to ensure it is reliable and does not have any bugs.

HARDWARE REQUIREMENTS:

- **Computer**: A modern computer with a dual-core processor, 4GB of RAM, and at least 20GB of available storage space.

- **Memory (RAM):** 4GB or more for efficient performance, depending on the project's scale.

- **Storage:** 20GB of available storage space for project files and SQLite databases.

- **Internet Connectivity:** An internet connection is required for installing packages, version control, and accessing online resources (if applicable).

- **Input and Output Devices:** A standard keyboard, mouse, and monitor for development and interaction with the GUI.

- **Backup System:** Regularly back up project files and database to prevent data loss.

SOFTWARE REQUIREMENTS:

- **Operating System:** Use any major operating system (Windows, macOS, Linux) for C development.

- **Development Environment:** Choose a text editor or a C-friendly IDE for writing and compiling your code.

- **Database:** Select a C-compatible database system, like SQLite or MySQL, for data storage.

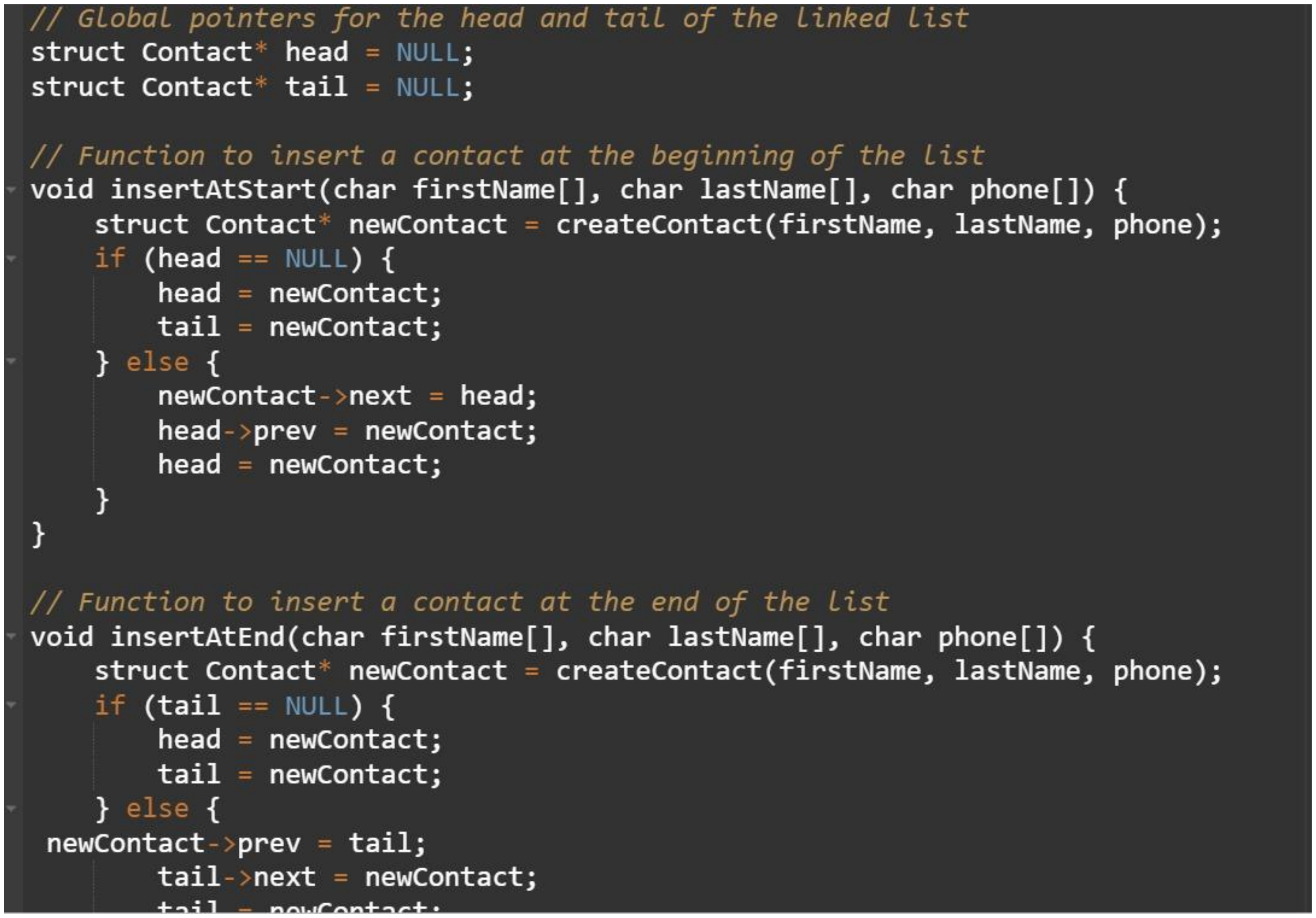
- **Version Control:** Employ Git for tracking code changes and collaboration.

- **Documentation Tools:** Document your code using plain text files or Markdown editors.

- **Testing Frameworks:** Create your testing functions or consider third-party C testing libraries like Check or Unity.

- **Dependency Management:** Manage external libraries manually since C lacks package managers like Python's pip.

USE OF DATA STRUCTURE



```
// Global pointers for the head and tail of the linked list
struct Contact* head = NULL;
struct Contact* tail = NULL;

// Function to insert a contact at the beginning of the list
void insertAtStart(char firstName[], char lastName[], char phone[]) {
    struct Contact* newContact = createContact(firstName, lastName, phone);
    if (head == NULL) {
        head = newContact;
        tail = newContact;
    } else {
        newContact->next = head;
        head->prev = newContact;
        head = newContact;
    }
}

// Function to insert a contact at the end of the list
void insertAtEnd(char firstName[], char lastName[], char phone[]) {
    struct Contact* newContact = createContact(firstName, lastName, phone);
    if (tail == NULL) {
        head = newContact;
        tail = newContact;
    } else {
 newContact->prev = tail;
        tail->next = newContact;
```

Fig. : This is a short part of our code in which insertion is being done.

In this project we have used mainly Double Linked List to implement Phone Directory. We have used double linked list to first create a linked list and then to insert in the beginning, at the end and at a specified position. We have also used data structure to display and sort the list in alphabetical order the list. We have again used double linked list to delete a contact from any position. By using traversal of linked list it is also possible to search for a contact by first name.

ADVANTAGES/DISADVANTAGES OF PHONE DIRECTORY

Advantages:

The phone directory app has a number of benefits.

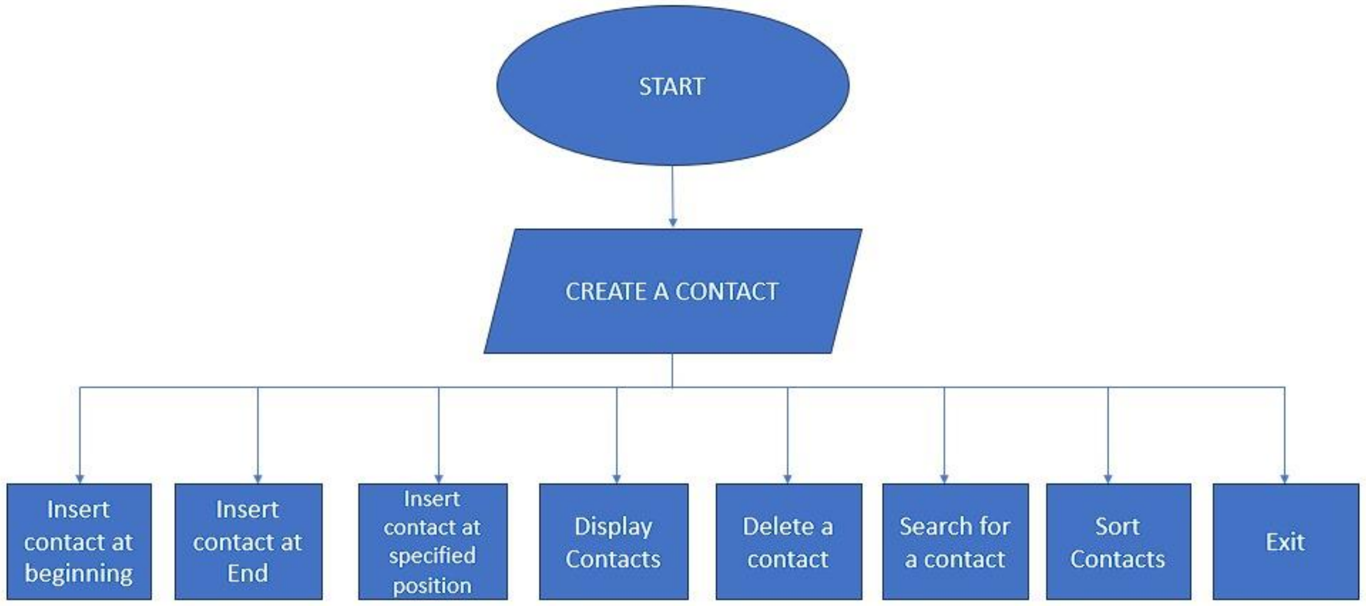
- ❖ User-Friendly Interface: Even those with just little computer knowledge may operate the menu-driven interface with ease.
- ❖ Contact Management: By adding, removing, and organizing contacts, users may effectively manage their contact lists.
- ❖ Teaching Tool: This code is meant to serve as a model for novices learning C programming and linked lists.
- ❖ Portable: The program is quite portable since it can operate on a variety of computing devices.

Disadvantages:

The phone directory application has certain restrictions in addition to its advantages.

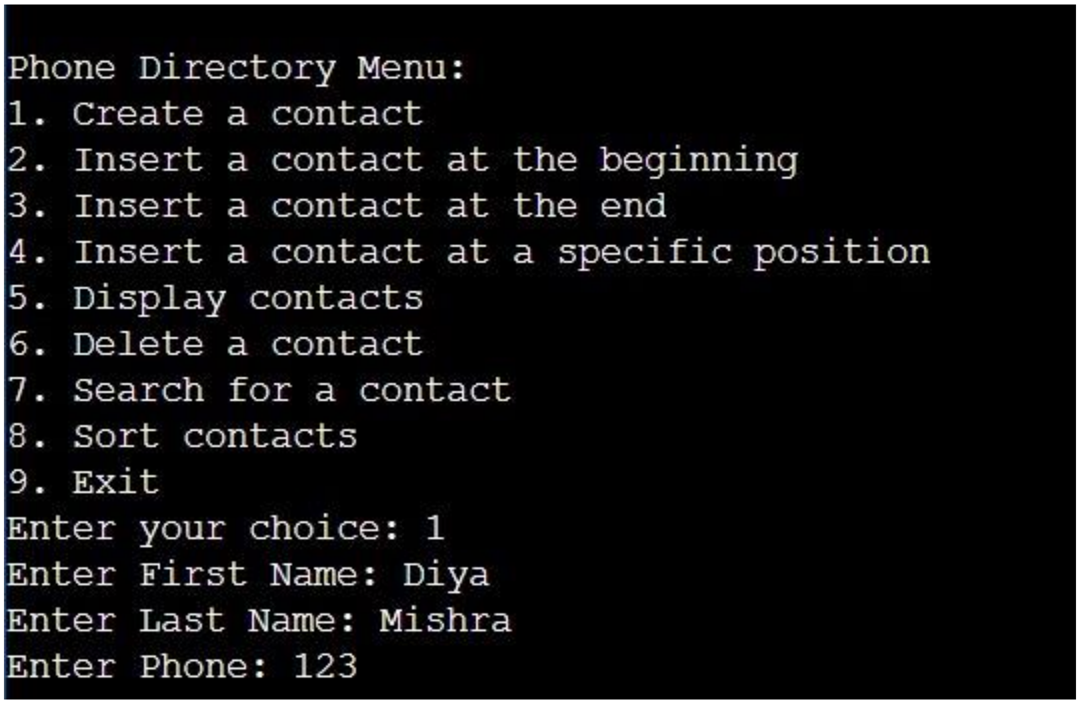
- ❖ Restricted Features: While commercial contact management software offers more complex functionality, this program only offers basic contact management features.
- ❖ Minimal Error Handling: The lack of thorough error handling for a number of instances in the code may occasionally result in unexpected behavior.
- ❖ Restricted Scalability: Being a console-based program, it might not be appropriate for handling large contact lists.

BLOCK DIAGRAM

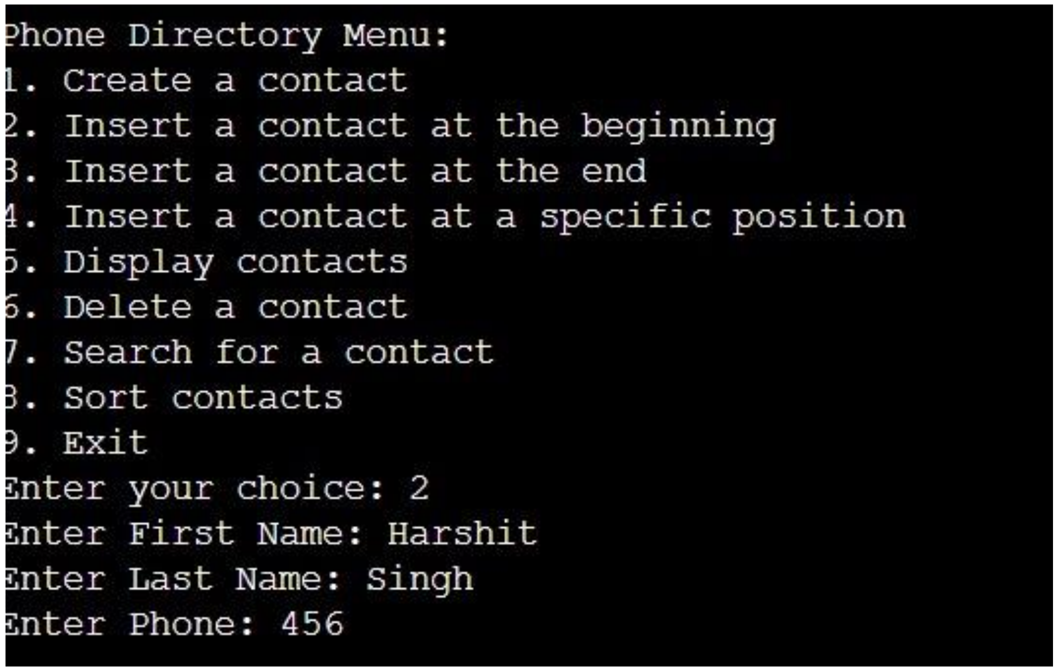


IMPLEMENTATION

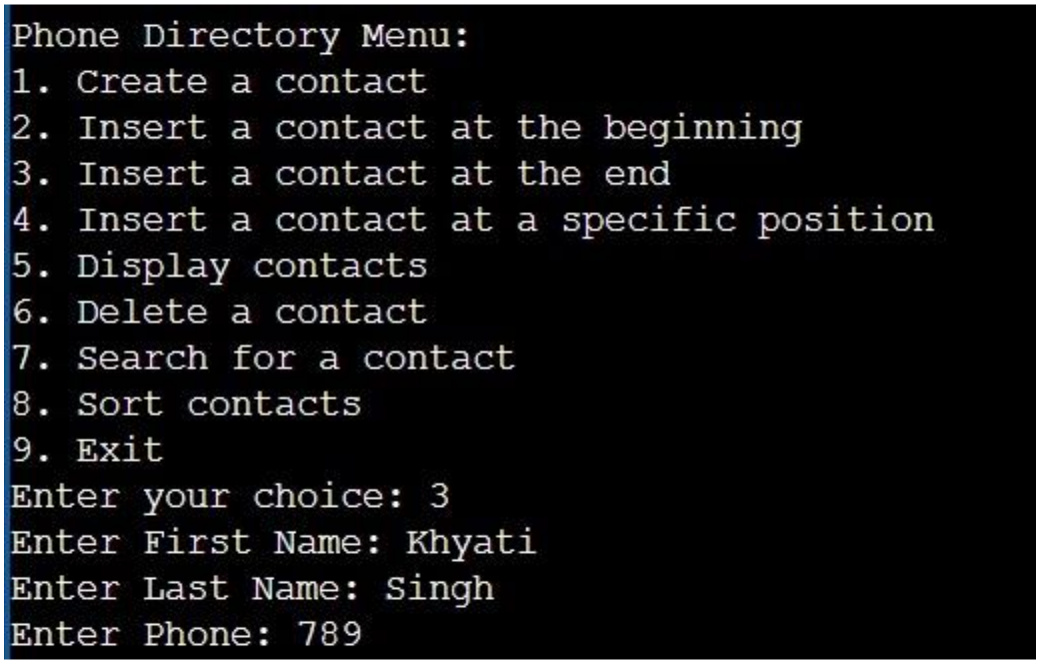
1. Creating a contact


```
Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 1
Enter First Name: Diya
Enter Last Name: Mishra
Enter Phone: 123
```

2. Inserting a contact at beginning

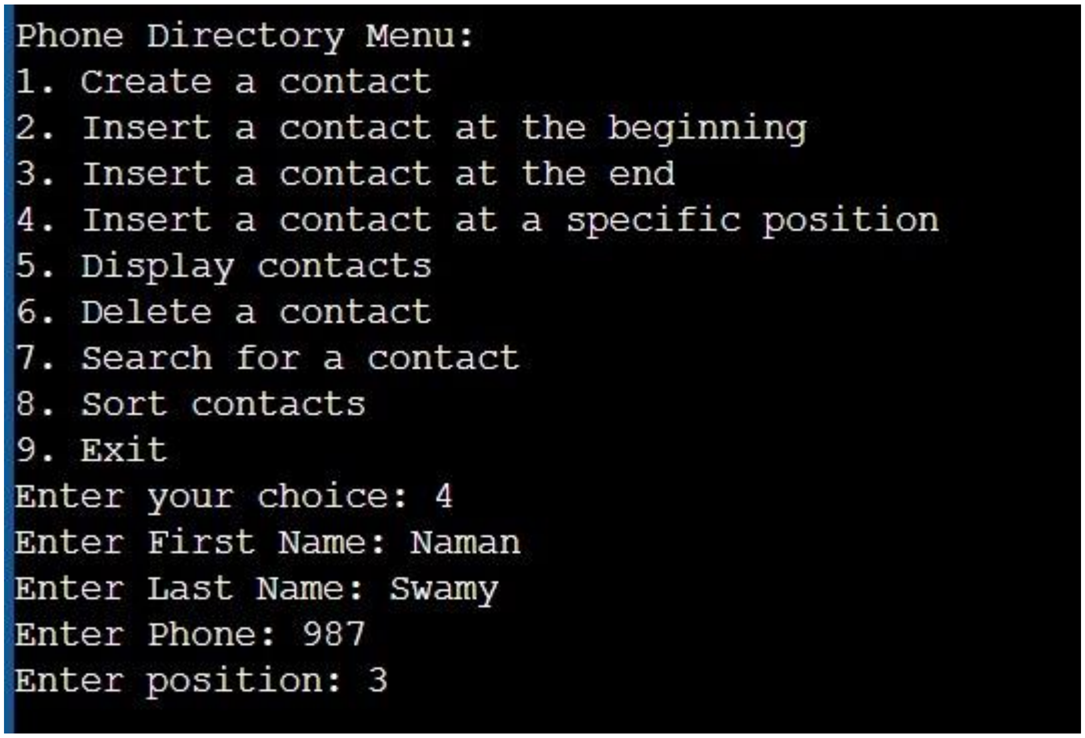

```
Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 2
Enter First Name: Harshit
Enter Last Name: Singh
Enter Phone: 456
```

3. Inserting at end



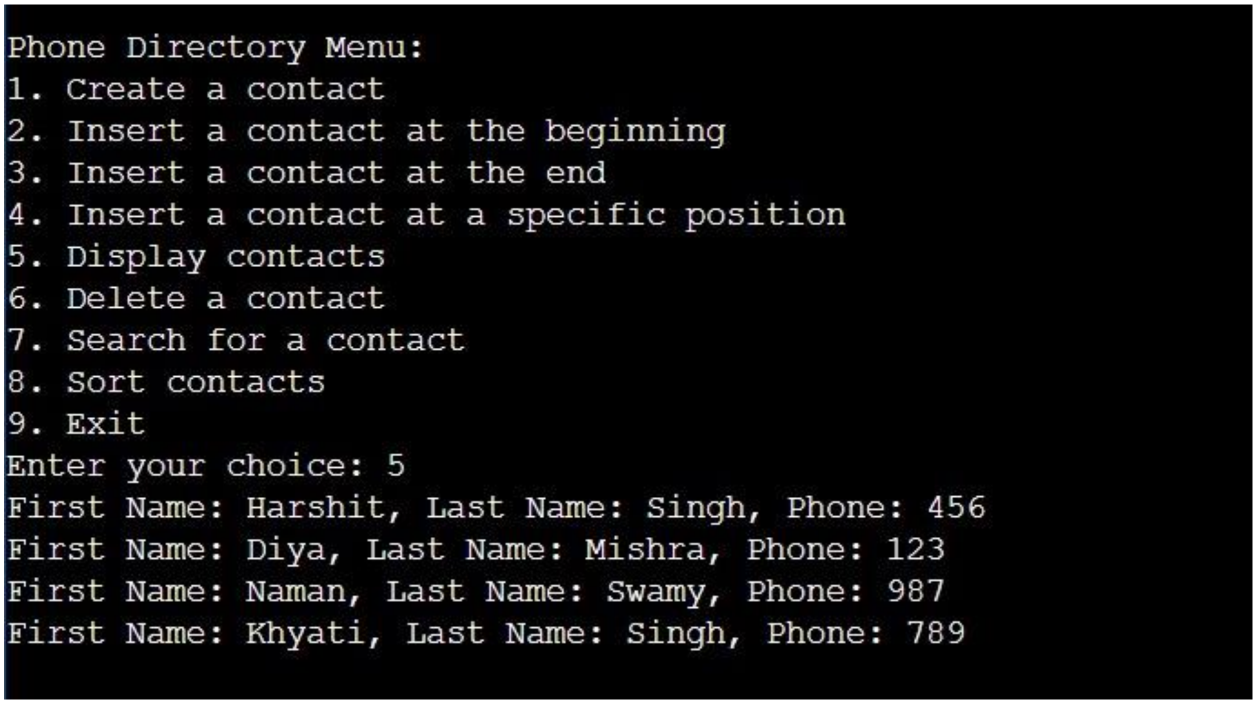
```
Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 3
Enter First Name: Khyati
Enter Last Name: Singh
Enter Phone: 789
```

4. Inserting at the position specified by the user



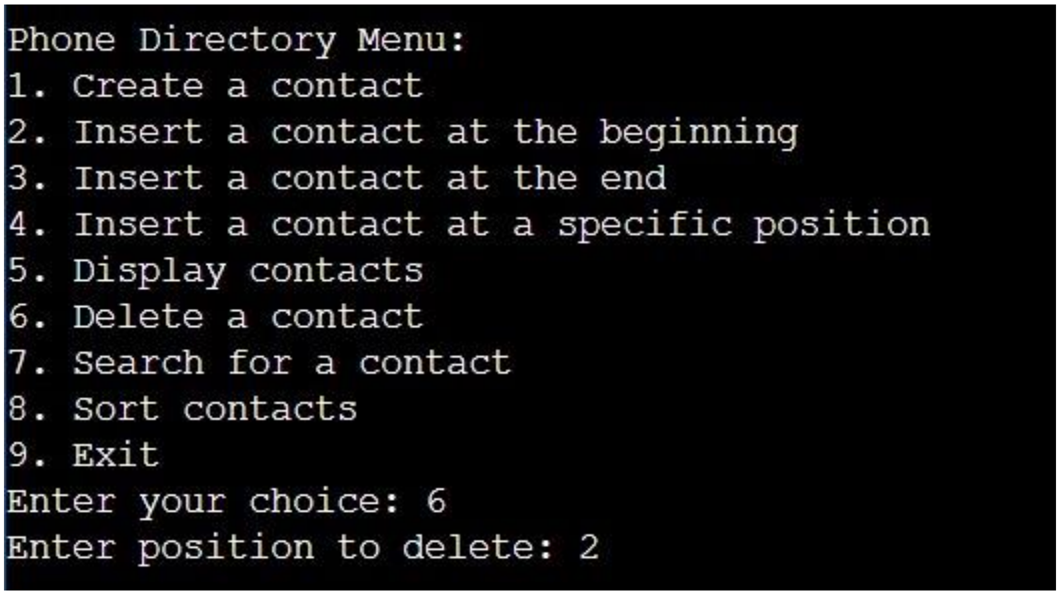
```
Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 4
Enter First Name: Naman
Enter Last Name: Swamy
Enter Phone: 987
Enter position: 3
```

5. Displaying the list of contacts



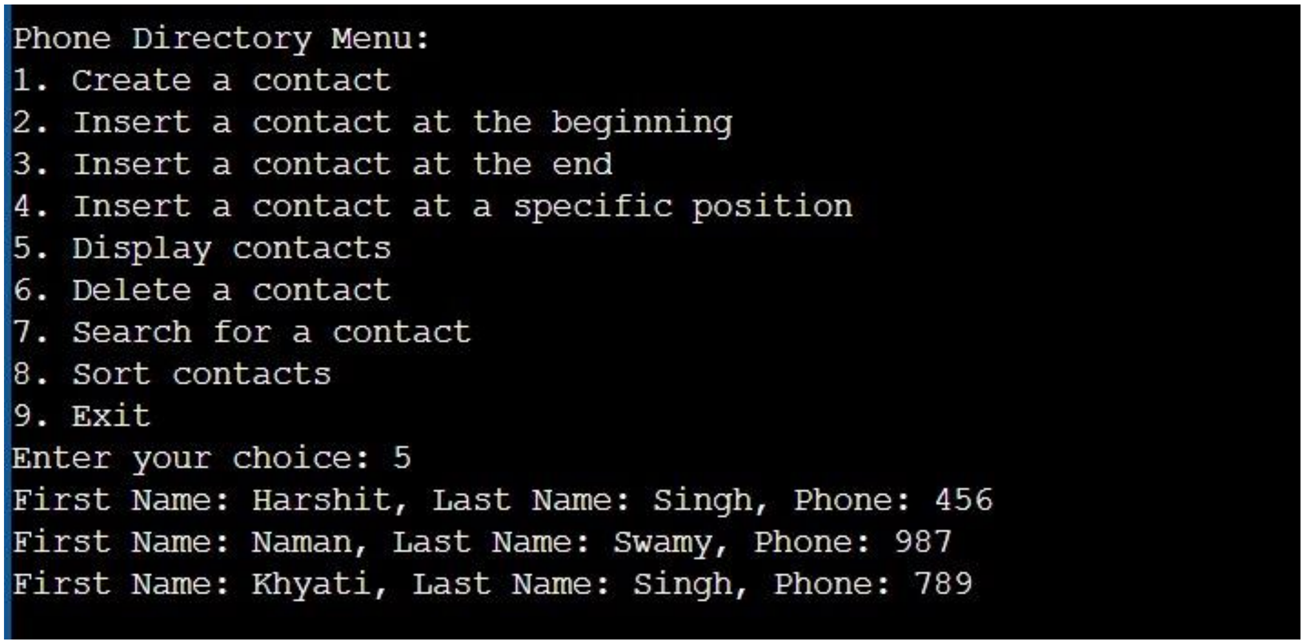
```
Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 5
First Name: Harshit, Last Name: Singh, Phone: 456
First Name: Diya, Last Name: Mishra, Phone: 123
First Name: Naman, Last Name: Swamy, Phone: 987
First Name: Khyati, Last Name: Singh, Phone: 789
```

6. Deleting a contact



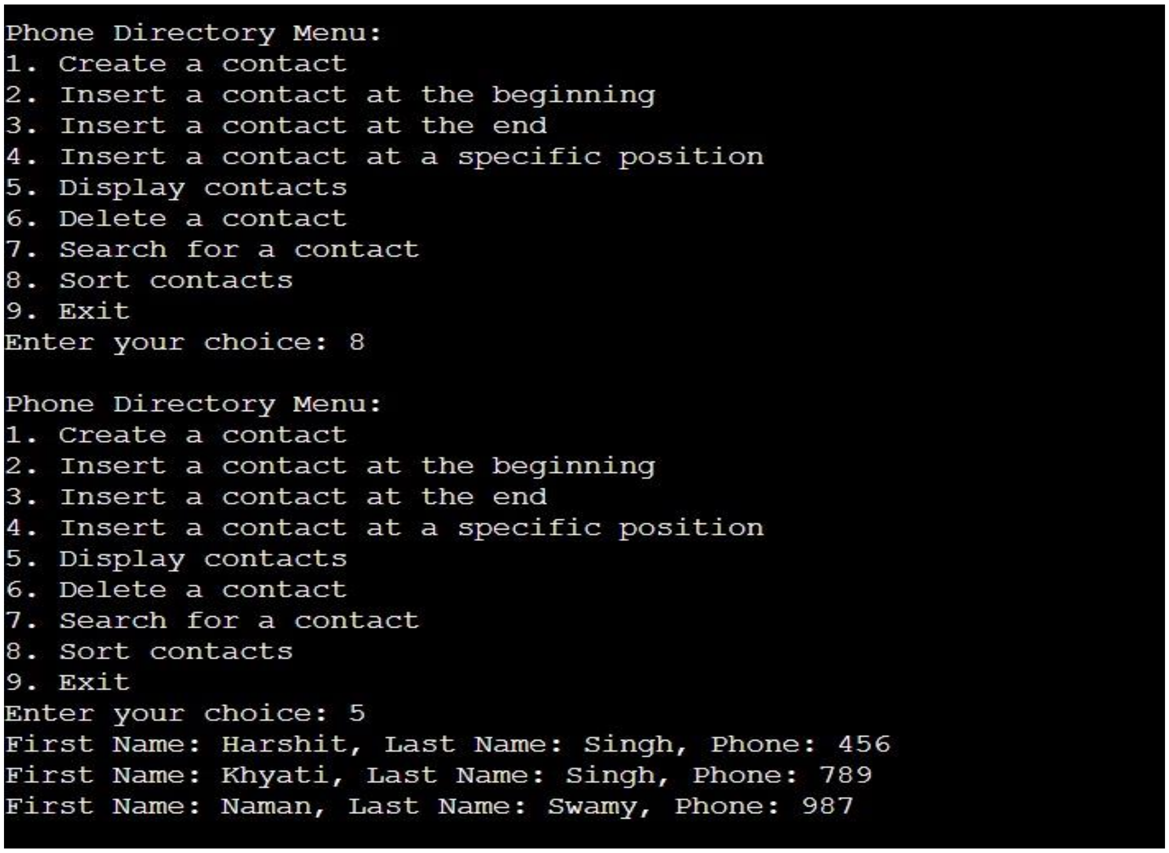
```
Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 6
Enter position to delete: 2
```

7. Displaying the updated list



```
Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 5
First Name: Harshit, Last Name: Singh, Phone: 456
First Name: Naman, Last Name: Swamy, Phone: 987
First Name: Khyati, Last Name: Singh, Phone: 789
```

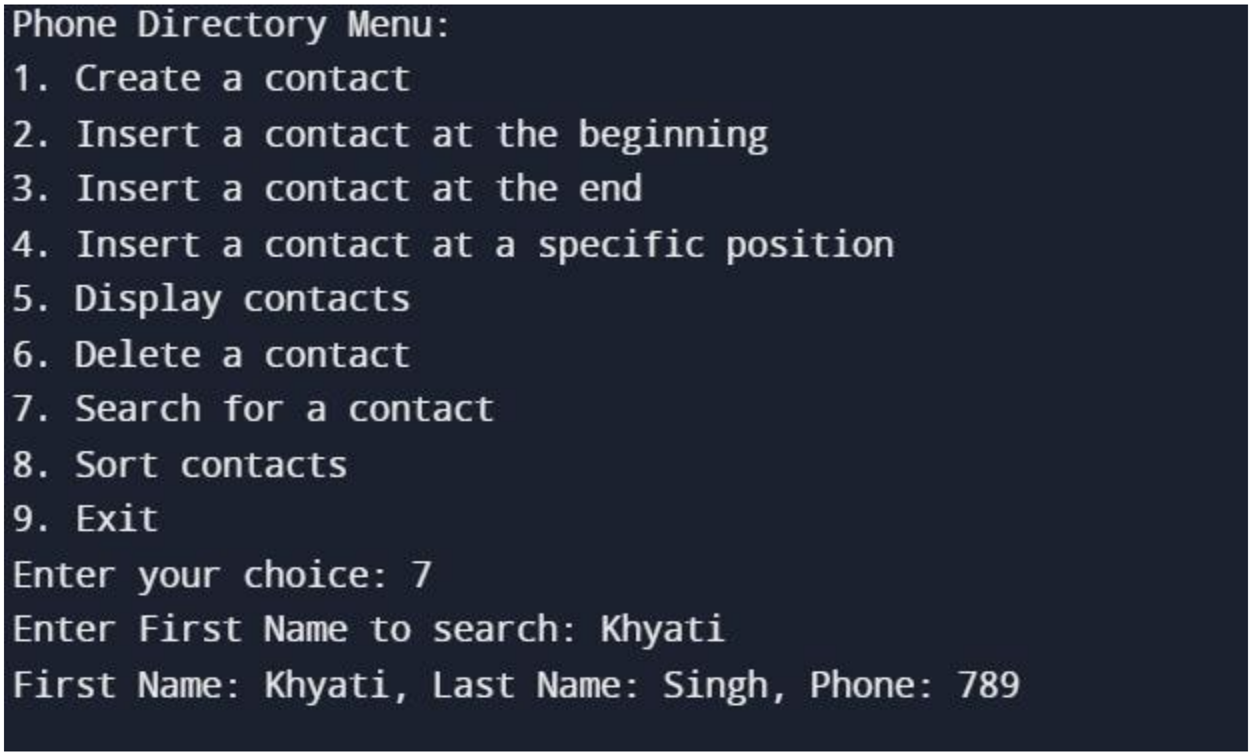
8. Sorting the contacts in an alphabetical order



```
Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 8

Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 5
First Name: Harshit, Last Name: Singh, Phone: 456
First Name: Khyati, Last Name: Singh, Phone: 789
First Name: Naman, Last Name: Swamy, Phone: 987
```

9. Searching a contact



```
Phone Directory Menu:
1. Create a contact
2. Insert a contact at the beginning
3. Insert a contact at the end
4. Insert a contact at a specific position
5. Display contacts
6. Delete a contact
7. Search for a contact
8. Sort contacts
9. Exit
Enter your choice: 7
Enter First Name to search: Khyati
First Name: Khyati, Last Name: Singh, Phone: 789
```

CONCLUSION

The project successfully implements a phone book management system using a linked list data structure in C. It allows user to insert, display, search, delete contacts, and exit the program. It utilizes a linked list data structure to store contact information (name and phone number) and provides a simple command-line interface for interaction. The system accomplishes the essential tasks of contact management. It demonstrates a clear understanding of data structures, memory management, and user interaction in a console-based application.