

Termi Contact

Contact Management System

Course : Algorithm and Computational thinking

Present by Am Vathanak & Chay Senghap

Lecturer : Dr. Va Hongly

Date : 18 - July - 2025

Background Information for this project

What and why about this project ?!



present by Am Vathanak & Chay Senghap

Background Information for this project

- **Contact management systems are used to store and organize personal or professional contacts.**
- **We built a basic contact manager using the C programming language.**

Name	Phone number	Gmail
john	+1 415-555-1234	john13@gmail.com
Mary	090 23 23 23	mary2@gmail.com

What and why about this project ?!

- **What : TermiContact is a simple console-based contact management program.**
- **Why : To practice core C programming skills like struct, file handling, and input validation, and to simulate a real-world application.**

Key feature about my Contact management

1. User Guidline

```
===== [ TermiContact User Guideline ] =====

This app lets you manage your contacts easily through a simple menu system.

1.Add New Contact      -> Input name, phone number, and email. No duplicates.
2.Edit Contact          -> Change a contact's name, phone number, or email.
3.Show All Contacts     -> Displays all saved contacts in a list format.
4.Search for Contact    -> Find a contact by typing its name.
5.Delete Contact         -> Remove a contact by name.
6.Delete All Contacts   -> WARNING: Permanently deletes all saved contacts.
7.User Guideline         -> Shows this help page.
8.About Us               -> Information about the creator and contact links.
0.Exit                   -> Exit the application with confirmation.
```



Key feature about my Contact management

- 1. User Guildline**
- 2. Add New Contact**

Key feature about my Contact management

- 1. User Guildline**
- 2. Add New Contact**
- 3. Edit your Contact**

Key feature about my Contact management

- 1. User Guildline**
- 2. Add New Contact**
- 3. Edit your Contact**
- 4. Search For Contact**

Key feature about my Contact management

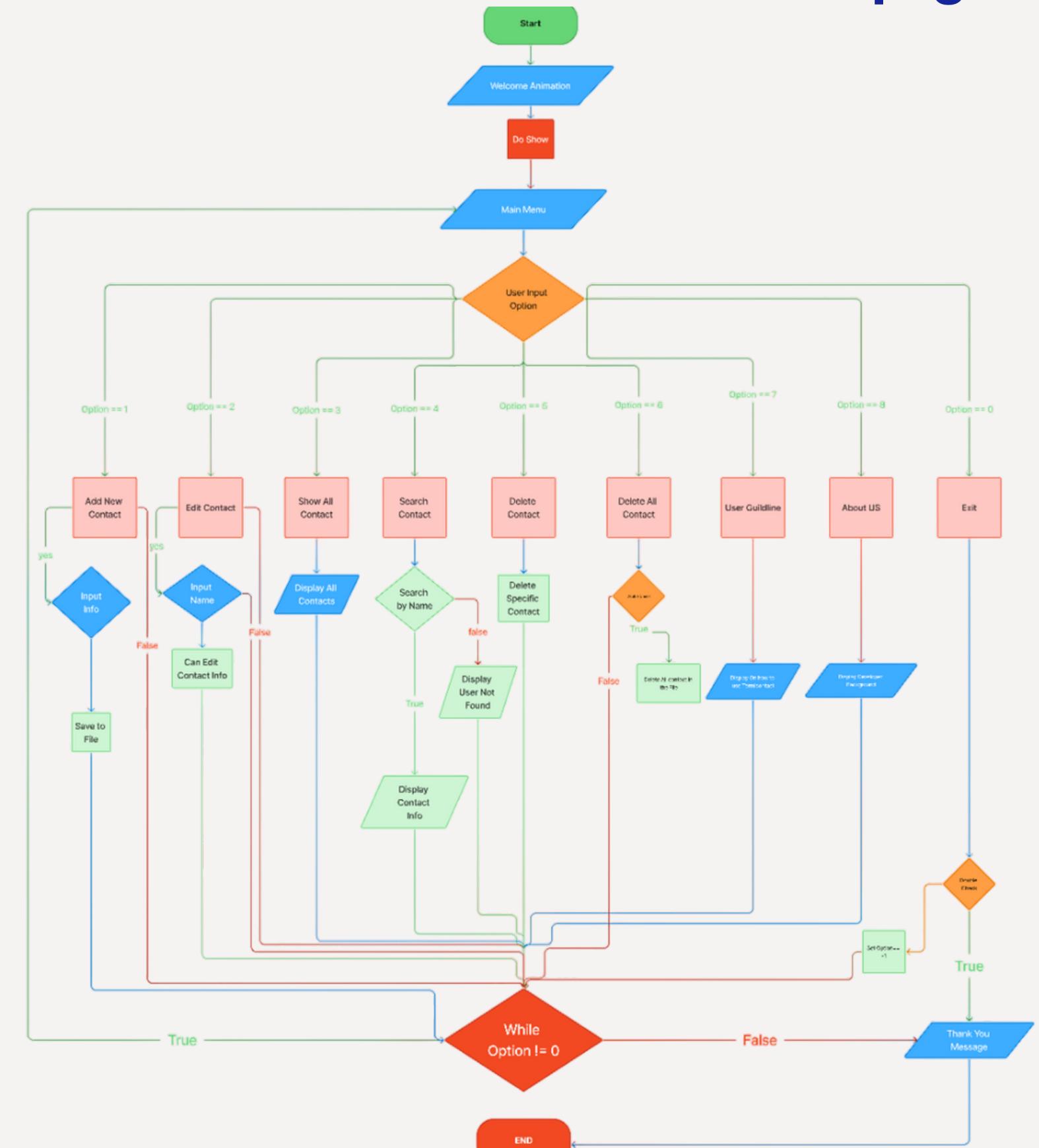
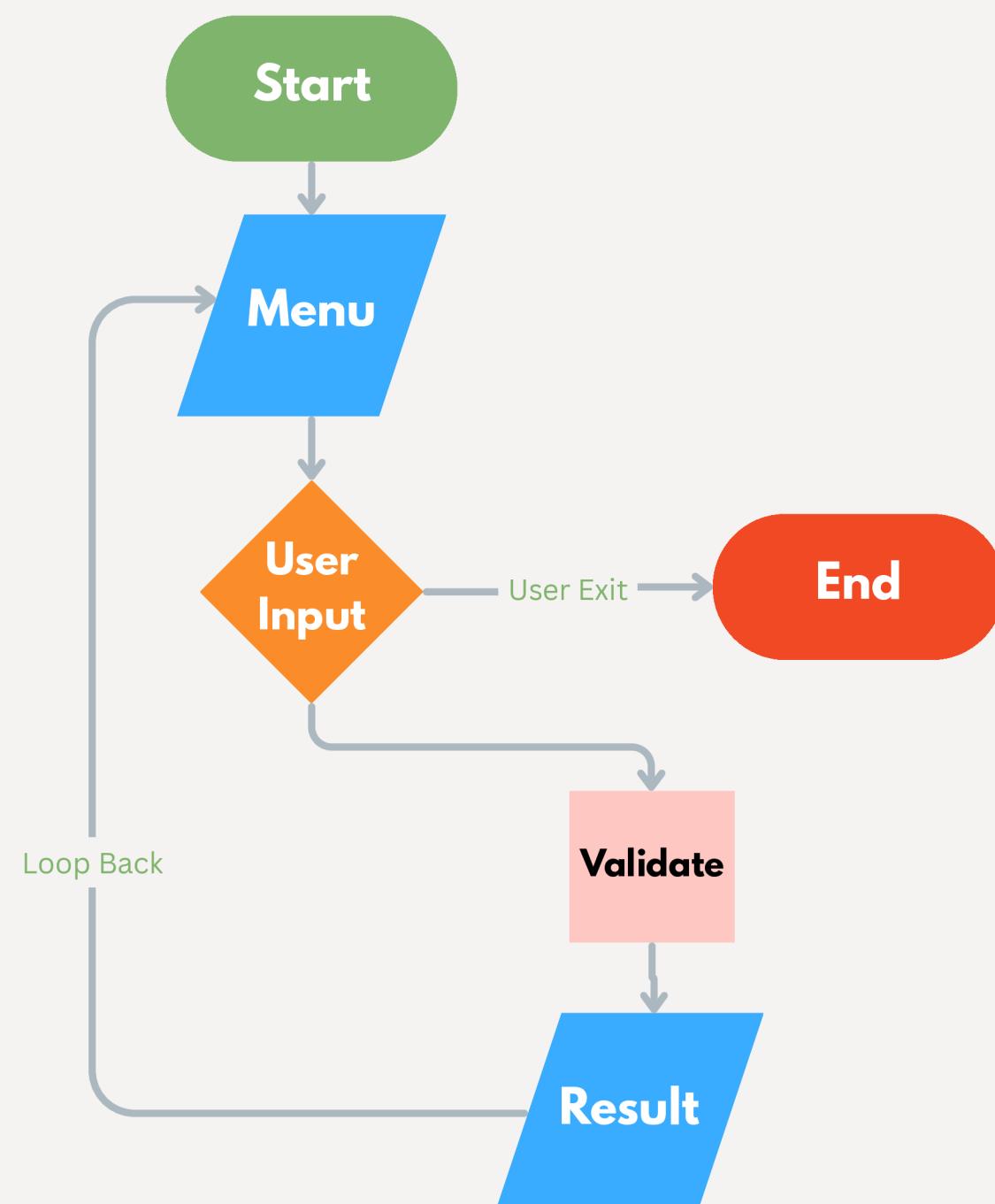
- 1. User Guildline**
- 2. Add New Contact**
- 3. Edit your Contact**
- 4. Search For Contact**
- 5. Delete Contact**

Key feature about my Contact management

- 1. User Guildline**
- 2. Add New Contact**
- 3. Edit your Contact**
- 4. Search For Contact**
- 5. Delete Contact**
- 6. About us**

Flowchart & Strategy

Quick Overview of the FlowChart



Result of the code

+ Program Execution Flow:

- **Welcoming Animation : Displays "Welcome To TermiContact..." with typing animation**
- **Main Menu Appears : Clean console interface with 8 options**
- **User Selection : Input validation ensures only 0-8 are accepted**
- **Feature Execution : Chosen function runs with loading animations**
- **Persistent Storage : All changes saved to Contact.txt automatically**



Test Cases:

1. Name

- Enter Valid Name



2. Phone Number

- Enter Valid Phone Number
- Phone Number Contain only Digits , " + " and " - "



IF (Input Existing Contact (Name , Phone Number && Email)) {will lead to a pop up message and let user Re-Input };

ELSE{

valid contact

};

3. Email

- Enter Valid Email that Contain @ and Dot “.”



Test Cases:

Invalid Input

```
===== [ Add New Contact ] =====  
Input Name : 123Hab  
Name cannot start with a number!  
Input Name : Alex Sander  
Input Phone Number : ZeroOneTwoThree  
Invalid Phone Number (only digits, + or - allowed)  
Input Phone Number : +855-123-123-123  
Input Email : Alexgmailcom  
Invalid Email Address (must contain @ and .)  
Input Email : Alex@gmail.com  
Successfully Added New Contact  
=====
```

Check For Existing Contact

Test Cases:

```
=====*[ Add New Contact ]=====

Input Name : Jonh
Name already exists in contacts!

Input Name : Alex Sander
Name already exists in contacts!

Input Name : Alice
Input Phone Number : 098765432
Phone number already exists in contacts!

Input Phone Number : 012345678
Input Email : Alex@gmail.com
Email address already exists in contacts!

Input Email : Alice@gmail.com
Email address already exists in contacts!

Input Email : example@gmail.com
Successfully Added New Contact
=====
```

**D
E
M
O**

```
=====  
===== TermiContact =====  
=====  
= MENU =  
= 1. Add New Contact =  
= 2. Edit Contact =  
= 3. Show All Contacts =  
= 4. Search Contact =  
= 5. Delete Contact =  
= 6. Delete All Contacts =  
= 7. User Guide =  
= 8. About Us =  
= 0. Exit =  
=====  
Enter an Option : 1
```

Summary:

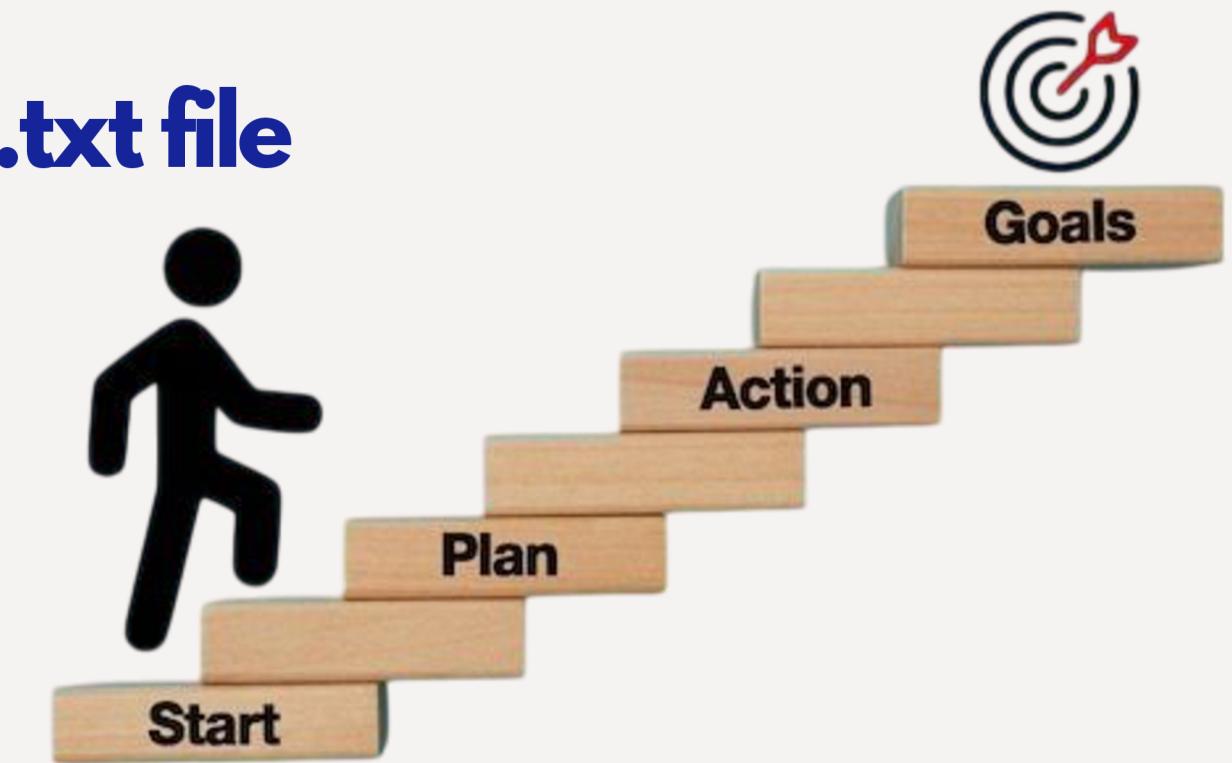
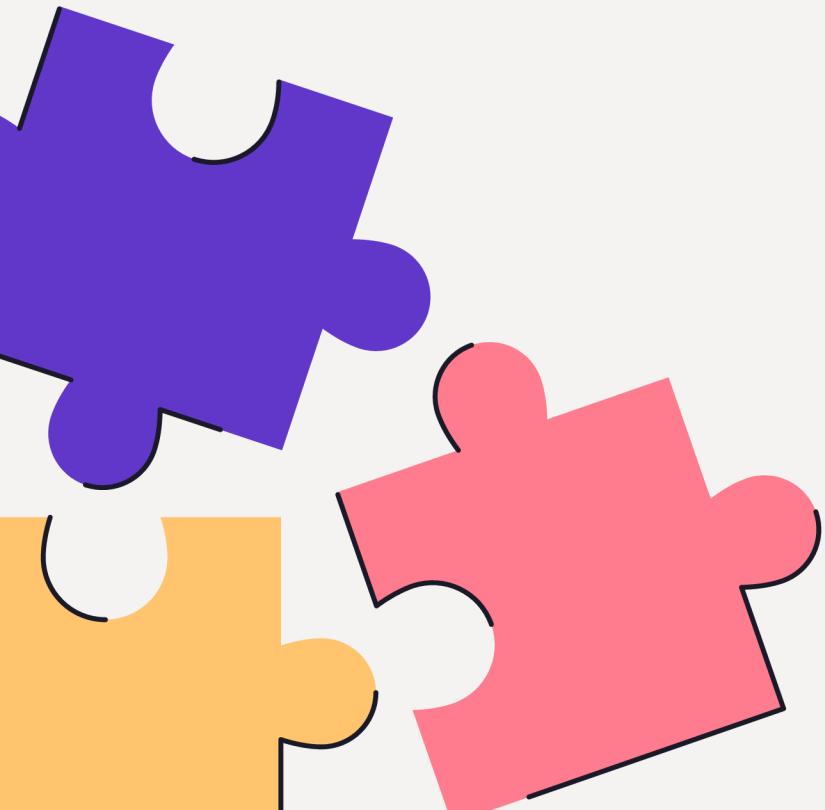
- **TermiContact is a console-based contact manager in C**
- **Allows users to add, edit, delete and store contacts in a text file**
- **Focused on applying real C programming features**



present by Am Vathanak & Chay Senghap

Key Achievements:

- Successfully added/edit contact data using file I/O
- Validated Everything User Input
- Contact data is stored persistently in Contact.txt file



present by Am Vathanak & Chay Senghap

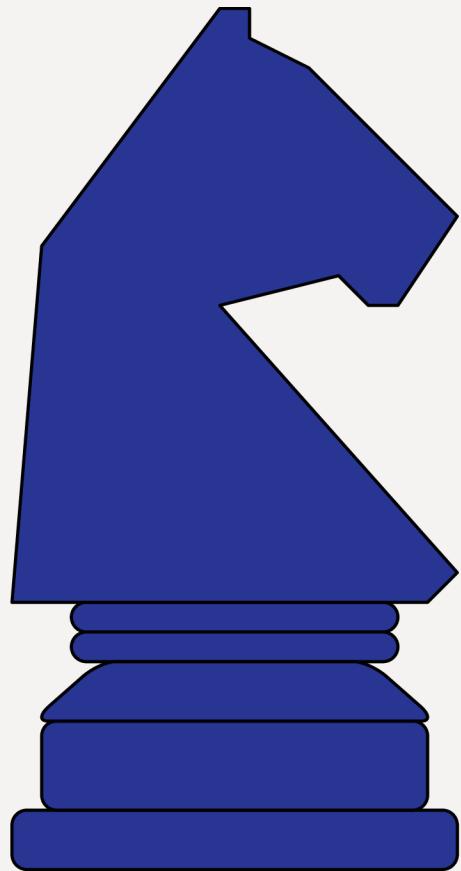
Challenges / Difficulties:

- Handling string input with fgets() and scanf()
- Avoiding data loss when editing files
- Validation Algorithm
- Fixing bugs while updating contact data



Lessons Learned:

- Improved understanding of **struct**, **FILE**, and **Validation Algorithm**
- Learned how to test and debug **step-by-step**
- Gained confidence in building a **complete C project**
- Team Work make a Dream Work



**Thank You
For listening**

