****

**Project Report**

**Programming in C-Lab**

**Title:** ACCOUNT MANAGEMENT SYSTEM

**Submitted By:-**

S*anchita Arora; 24070122283*

*Prachi Hassani; 24070122279*

*Sejal Jain; 24070122178*

*Sanjiti Gill; 24070122171*

**Under Guidance of**

***Dr. Rupali Gangarde***

**SYMBIOSIS INSTITUTE OF TECHNOLOGY**

**(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL UNIVERSITY)**

Pune

2024-25

**INTRODUCTION:-**

This is a simple account management system built in C, made specifically for macOS. It lets a user sign up, log in, recover forgotten credentials, change their password, or delete their account. Everything is done through the terminal with a clean menu-driven interface. The user’s information—like name, username, phone number, gender, DOB, and password—is stored in a file so it stays saved between runs.

A custom function is implemented to handle password entry by showing asterisks instead of the actual characters, making the process a bit more secure. While it's designed for managing one account at a time (each new registration overwrites the old data), it makes for a great beginner project to learn about file handling, working with structures, and handling user inputs in C.

**OBJECTIVES:-**

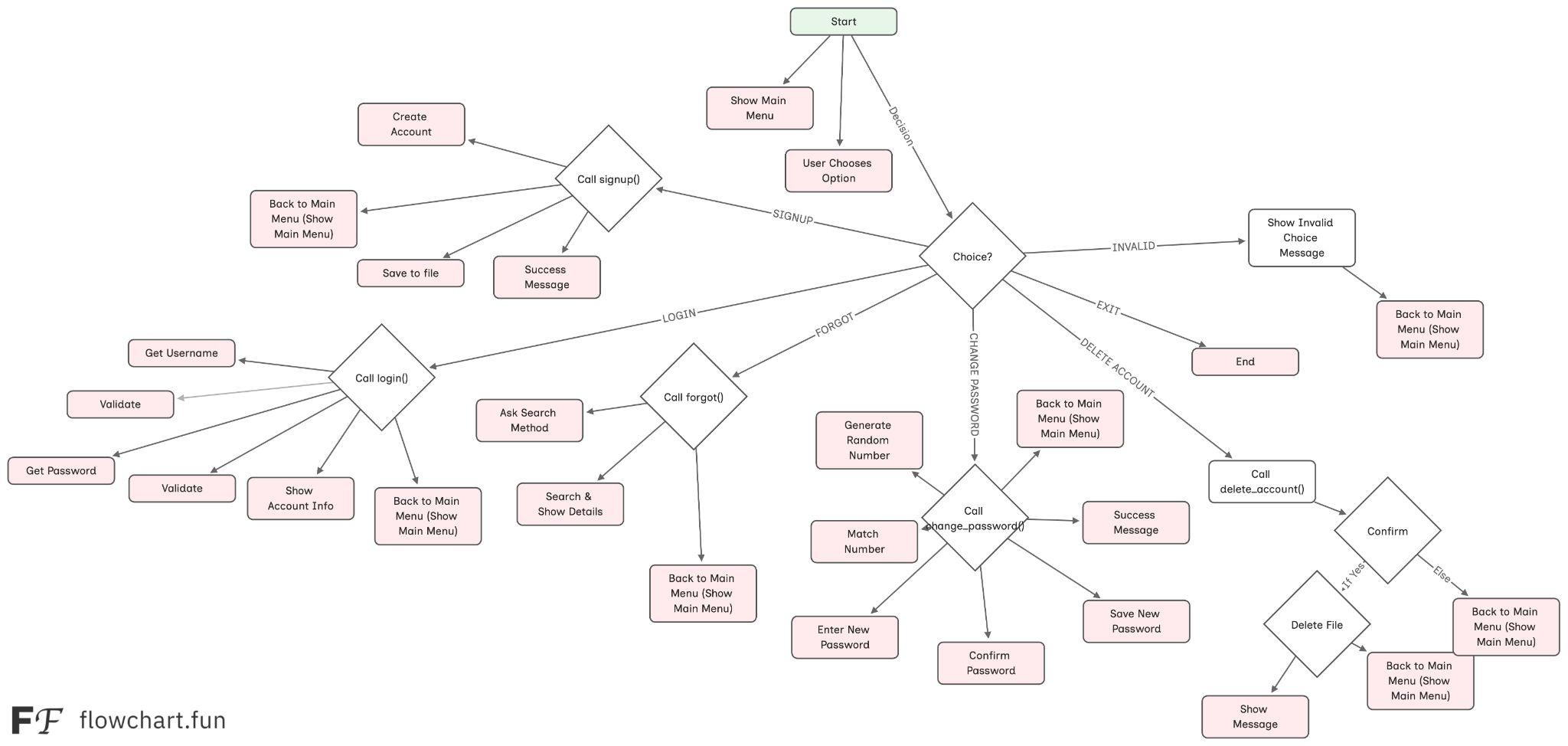
The objective section outlines the main goals and learning outcomes of this project. It highlights what this program is meant to achieve and what the user or developer can expect to learn or implement by working on it. These points help give a clear idea of the project’s purpose and the specific features it is designed to include.

* To create a basic account management system using C.
* To let users sign up by entering personal details and setting a password.
* To allow secure login using username and password.
* To include options for recovering forgotten usernames or passwords.
* To provide a way for users to change their password.
* To add an option for deleting an account if needed.
* To store user data in a file so it's saved even after exiting the program.
* To use structures for organizing and managing user data efficiently.
* To practice file handling and user input in a real-world-style mini project.

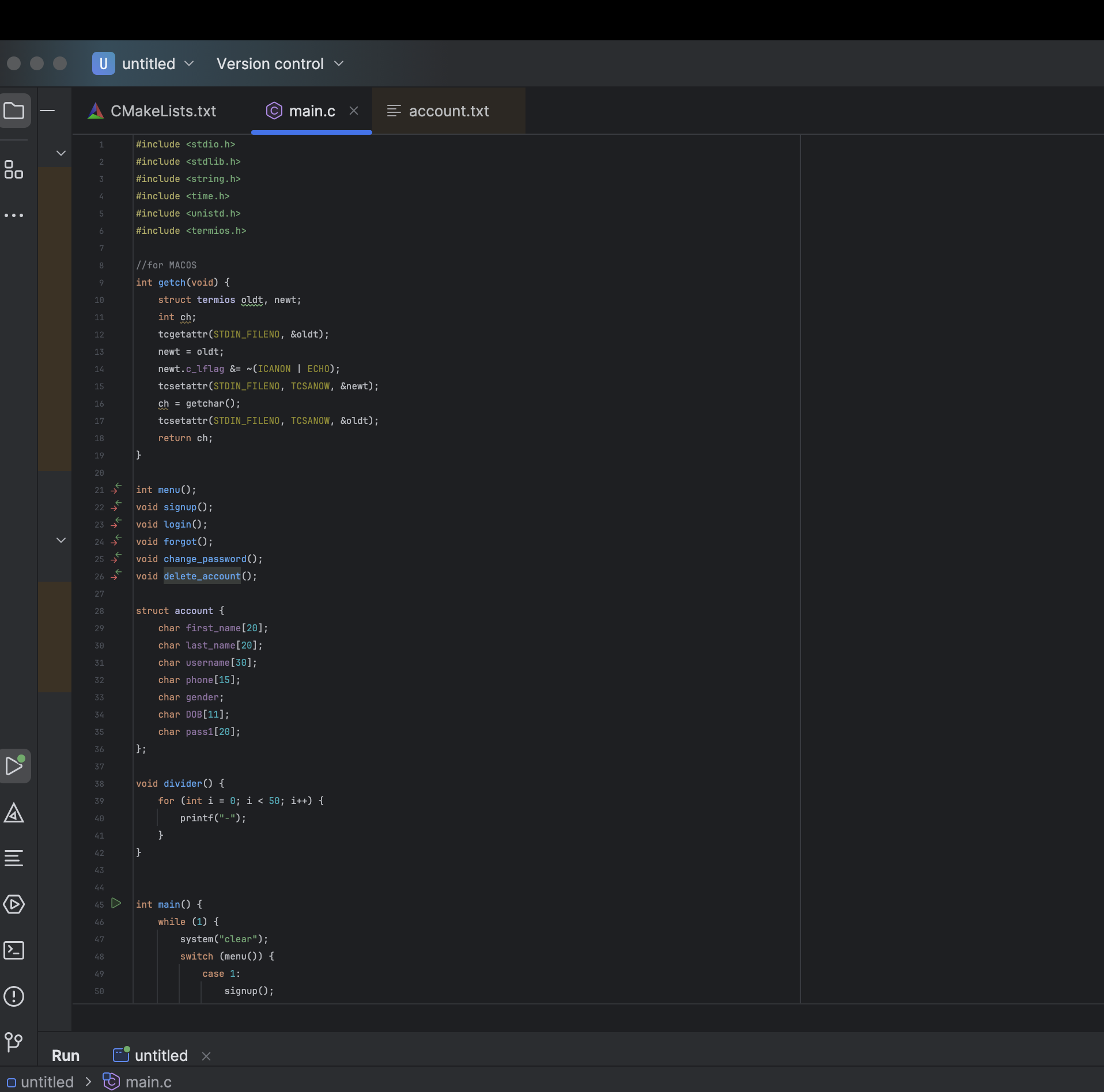
**ARCHITECTURE:-**

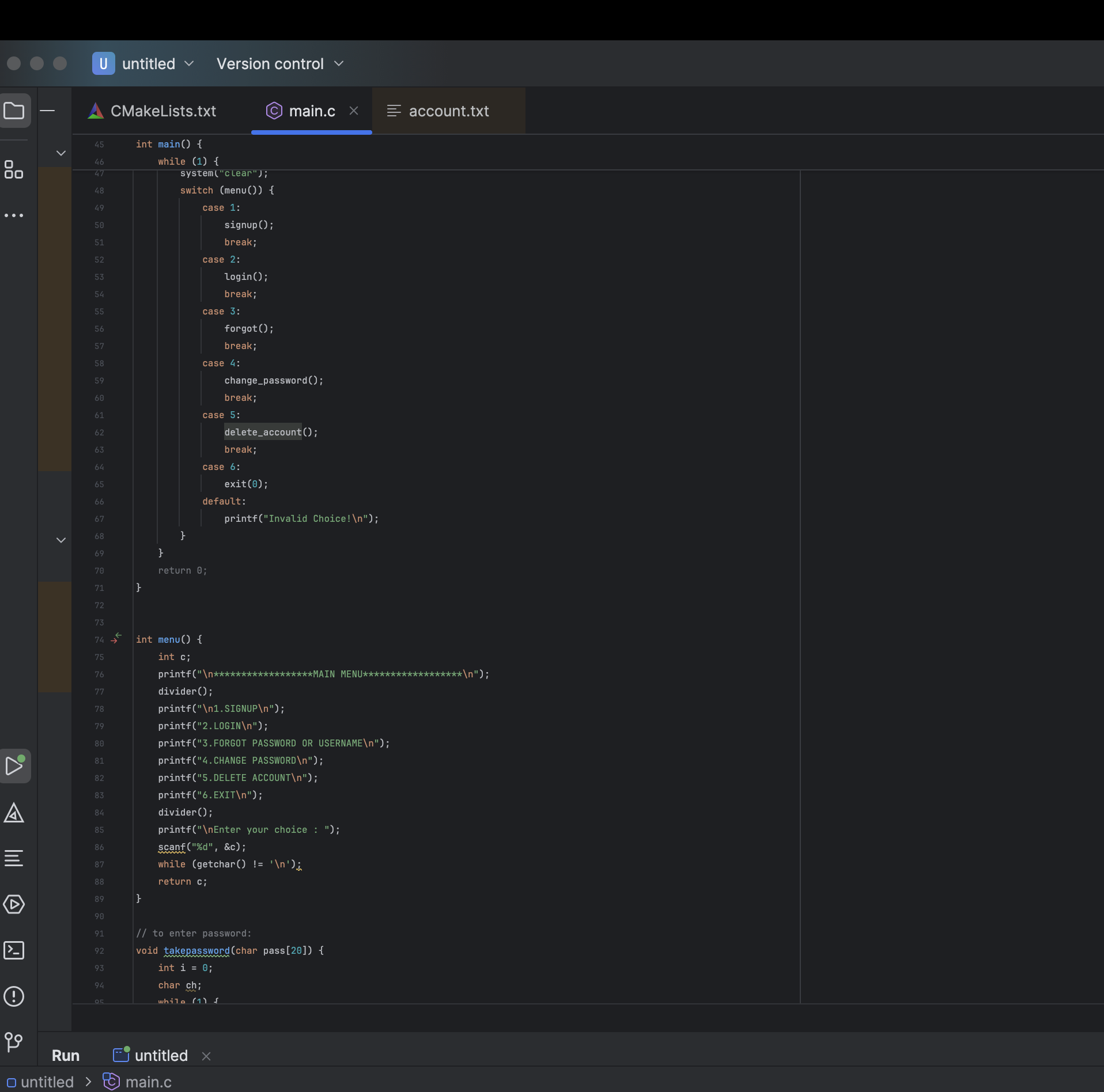
****

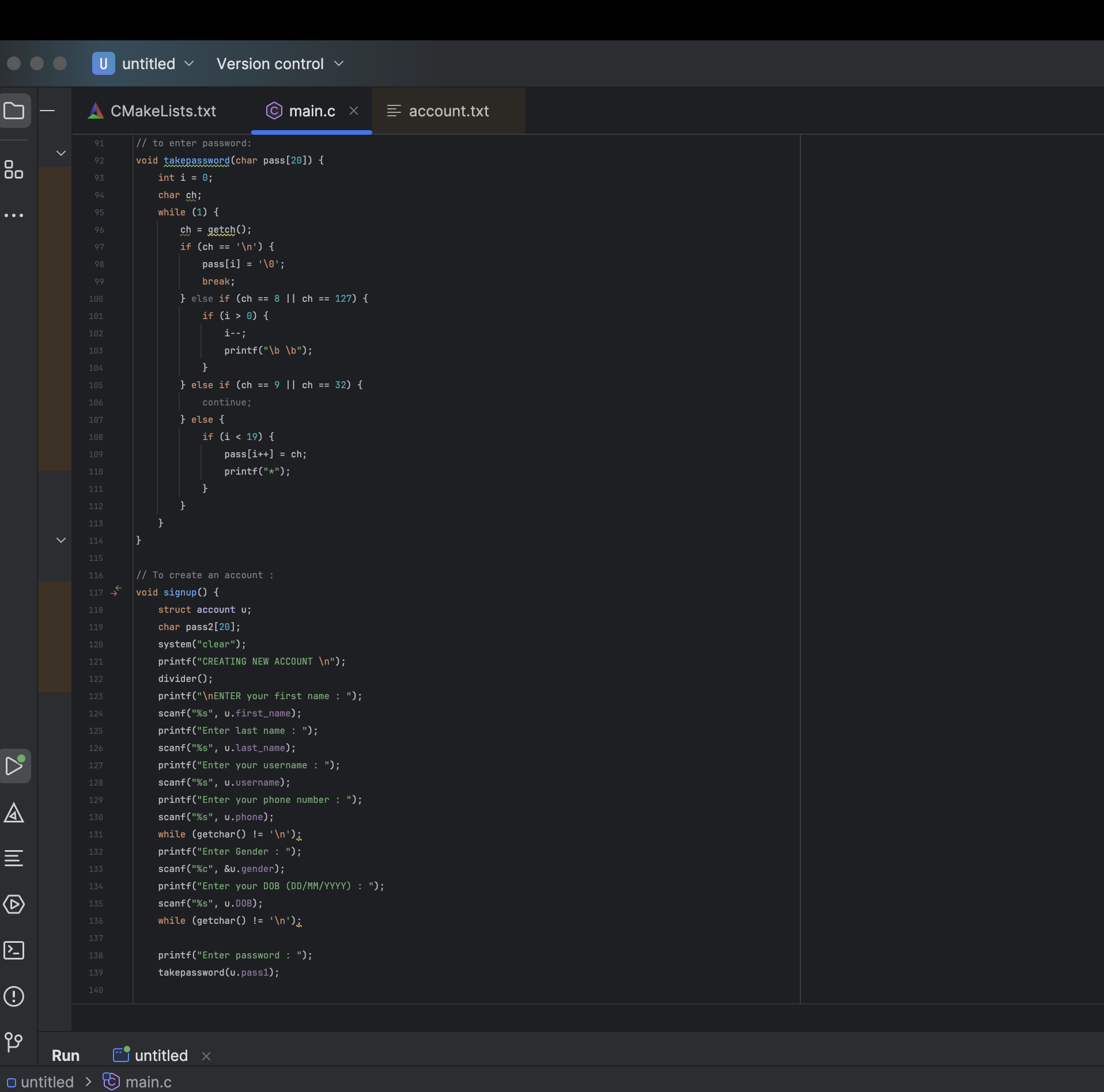
**FLOWCHART:-**

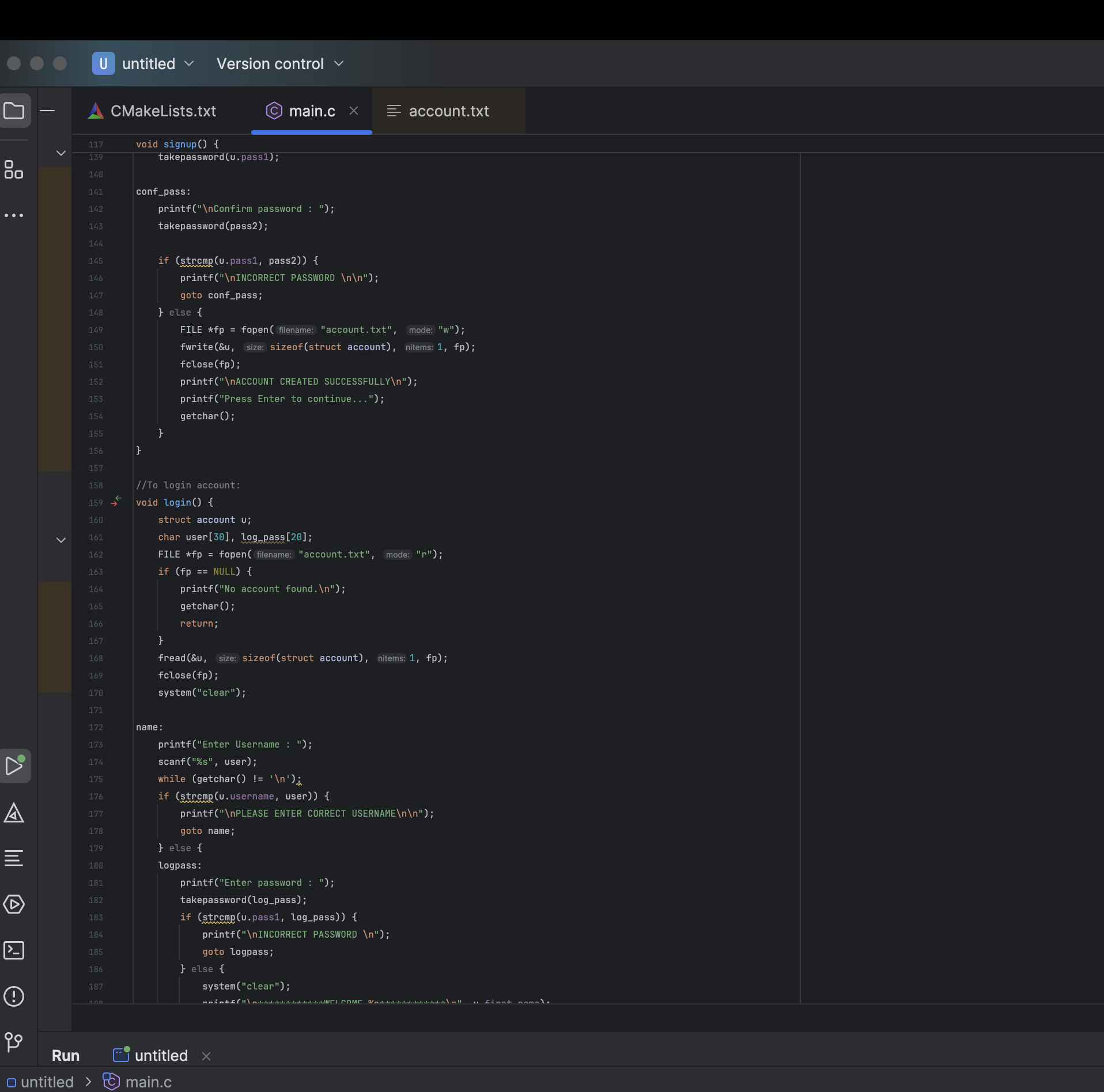
****

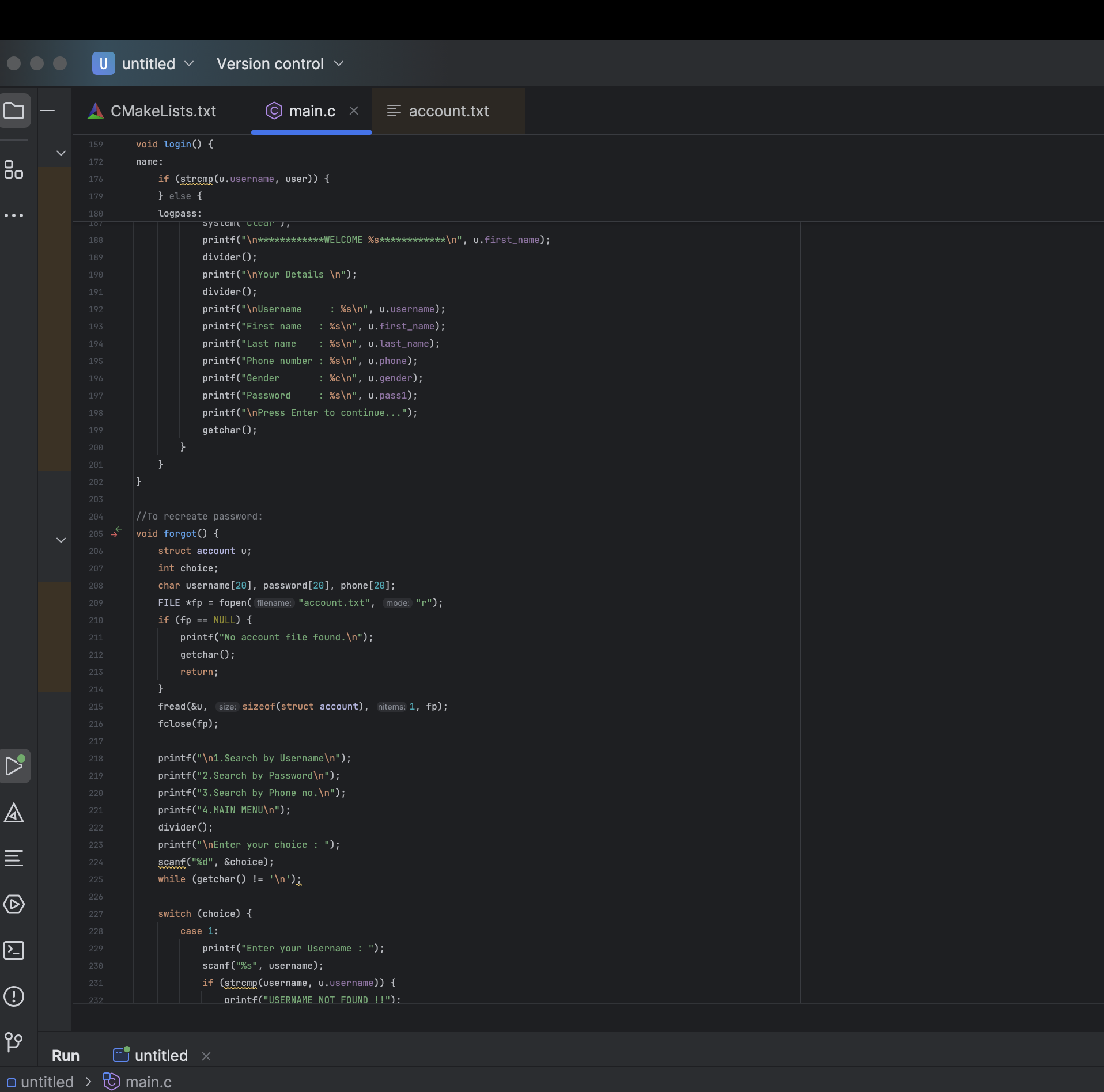
**CODE:-**

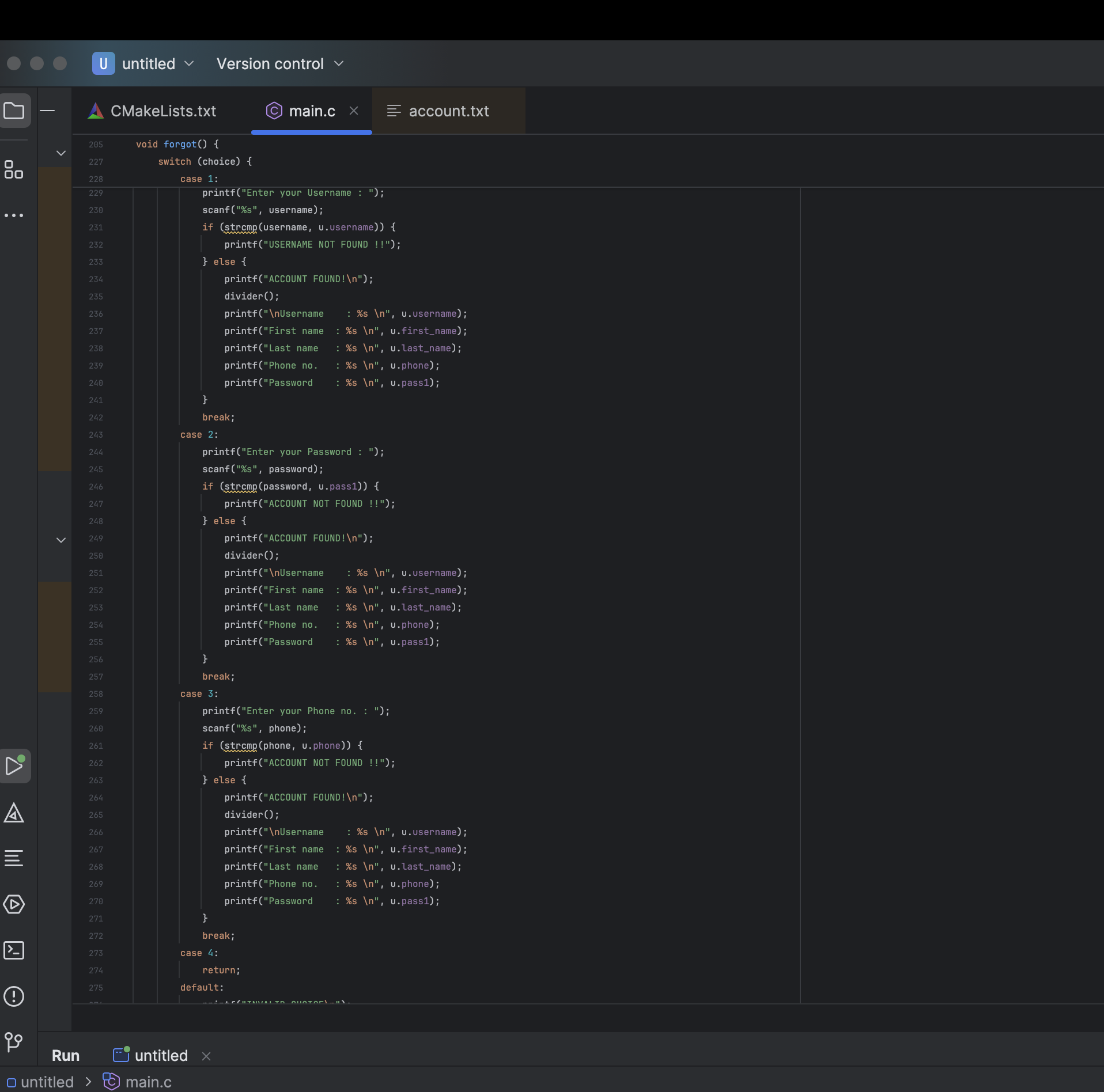
****

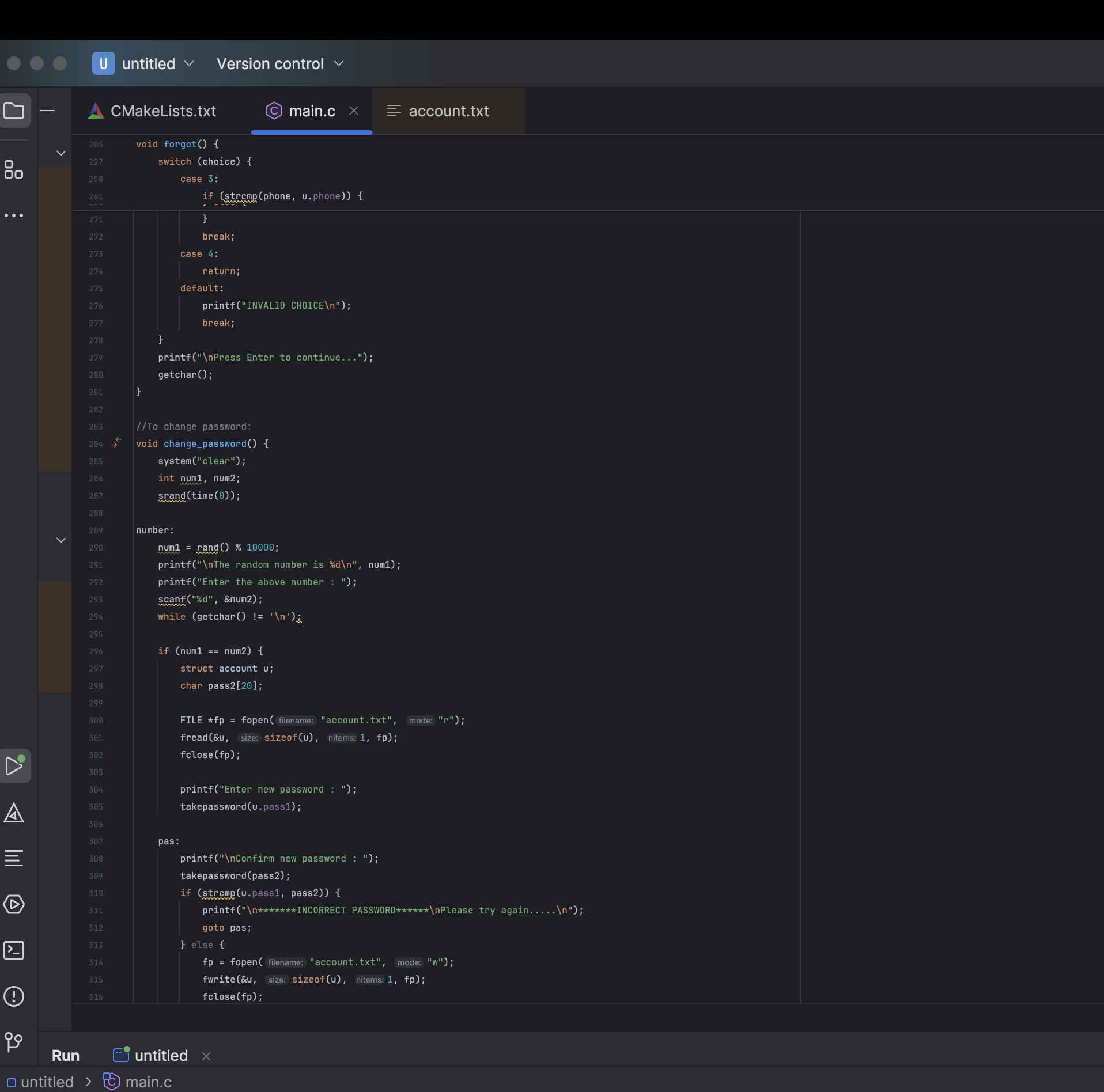
****

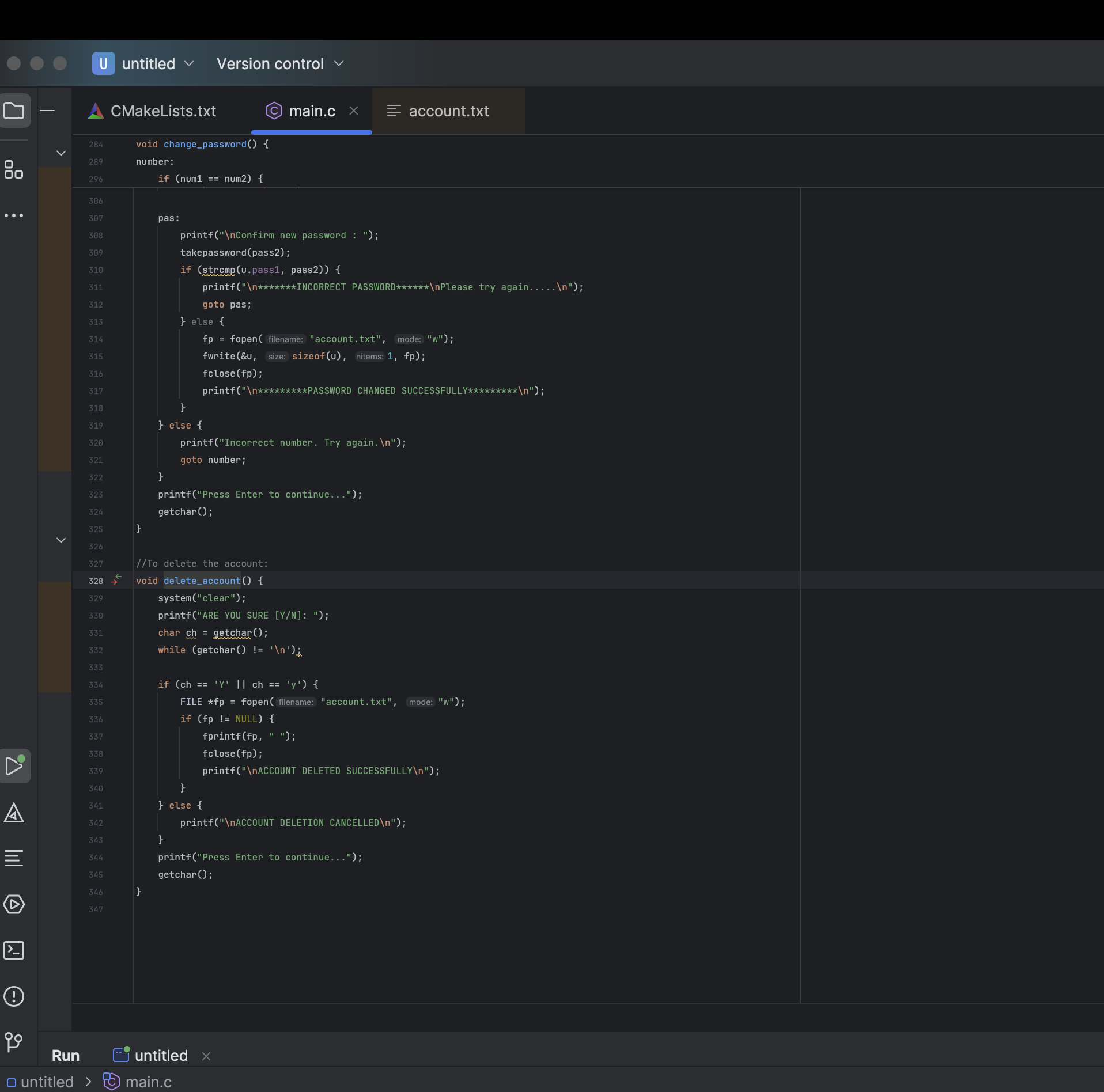
****

****

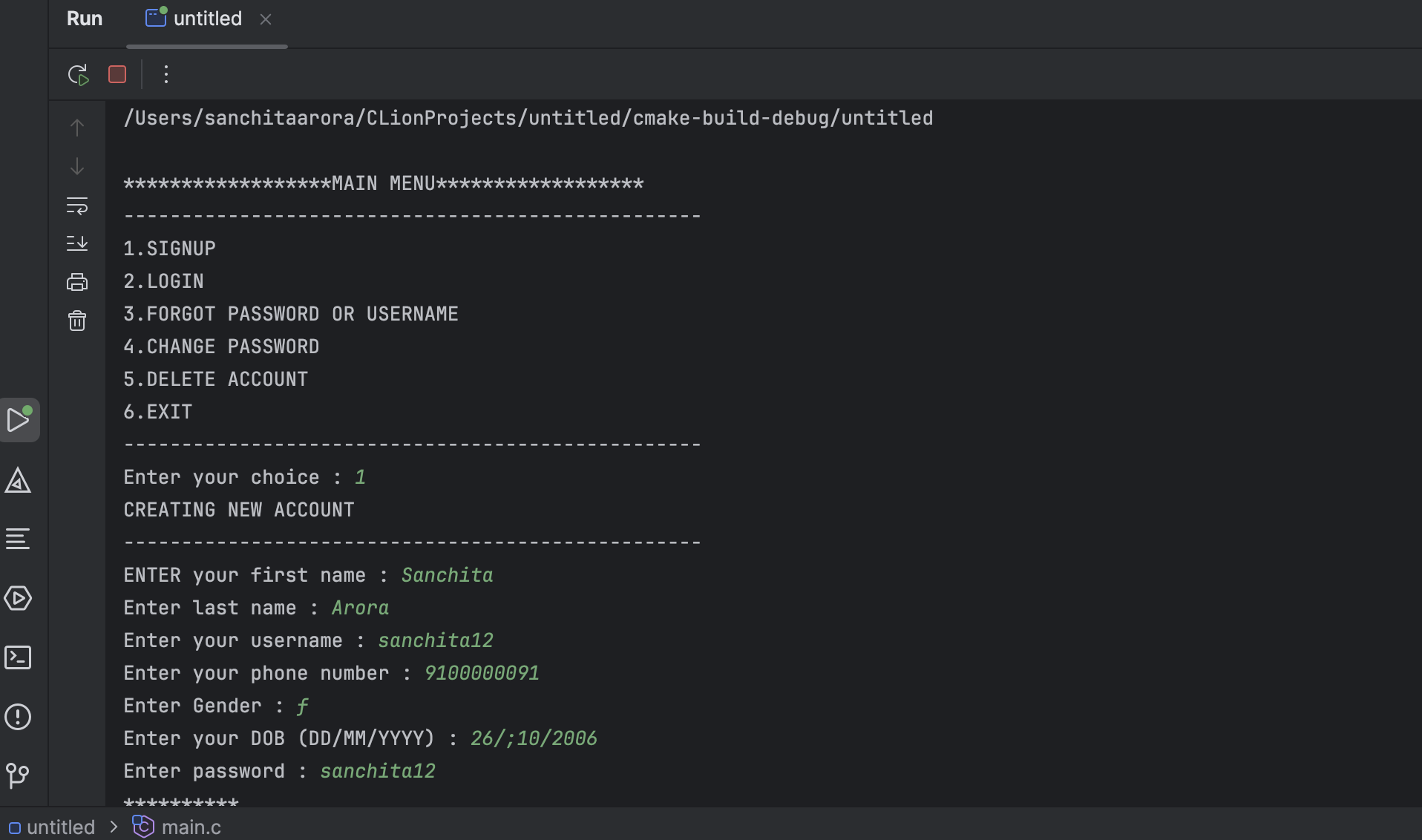
****

****

****

****

**OUTPUT:-**

****