Objective:

The objective of the project was to develop a simple Contact Book application in Python that allows users to store, search, and display contact information. Each contact consists of a name and a phone number.

Functionality:

The Contact Book application provides the following core functionalities:

Add Contact:

The user can add new contacts to the Contact Book. The application checks if the contact already exists by comparing the contact name (case-insensitive). If the contact exists, the application notifies the user. Otherwise, the new contact is added successfully.

Search Contact:

The user can search for a contact by entering the name. The search is case-insensitive, and the application returns the contact's name and phone number if found. If the contact does not exist, the user is informed.

Display Contacts:

The user can view all stored contacts. If there are no contacts in the Contact Book, the application notifies the user that there are no contacts to display. If contacts are present, the application lists all the names and phone numbers.

Exit Application:

The user can exit the application at any time by selecting the appropriate menu option.

Continuous Operation:

After performing any operation (add, search, or display), the application prompts the user to decide whether they wish to continue with further operations. If the user enters 'y', the application returns to the main menu; otherwise, the application exits.

Implementation Details:

Data Structure:

The contacts are stored in a list where each contact is a tuple consisting of a name and a phone number.

Functions:

- **add_con(contacts, name, phone):** Adds a new contact to the Contact Book if it doesn’t already exist.

- **search_con(contacts, name):** Searches for a contact by name and displays the contact's information if found.
- **display_con(contacts):** Displays all contacts stored in the Contact Book.
- **main():** Manages the application's flow, presenting a menu for the user to select operations and handling the continuation of the program based on user input.

User Interaction:

The user interacts with the application through a command-line interface (CLI), selecting operations from a menu. Each operation is followed by a prompt asking the user if they wish to continue, enhancing the user experience by allowing multiple operations within the same session.

Conclusion:

The Contact Book application successfully meets the requirements of storing, searching, and displaying contacts with a simple and intuitive user interface. The use of conditional statements, loops, and string handling ensures robust functionality. The addition of the continuous operation feature allows users to perform multiple actions in a single session, making the application user-friendly. This project serves as a basic yet functional example of Python programming with real-world applications.