Contact Book Application

The Contact Book app is created to give users a strong resource for organising their contacts efficiently. Its main goal is to provide capabilities like saving, listing, removing, finding, and sorting contacts. These abilities are designed to improve the ease of access to contact details.

Execution Instructions

1. Setup: Before running the Contact Book application, ensure Python is installed on your system. Download or clone the application code from the provided repository.
2. Running the Application:
    a. Clone the GitHub repository.
    b. Open a Jupyter Notebook environment on your local machine or a cloud-based platform like Google Colab, Anaconda Cloud etc.
    c. Import the Jupyter Notebook.
    d. Execute the notebook cells to interact with the Contact Book.

Design of Data Structure and Algorithms

Data Structures

1. Contact Information: Each contact is stored as a dictionary with the following keys:

    a. ID: A string representing the contact's unique identifier.

    b. name: A string representing the contact's name.

    c. phone: A string representing the contact's phone number.

2. Contact Book: The contact book itself is a list of dictionaries, where each dictionary represents a contact.

Algorithms

1. **Insertion**:

    a. Data Structure: A new contact is added to the list of contacts.

    b. Algorithm:

        i. Input: name, phone

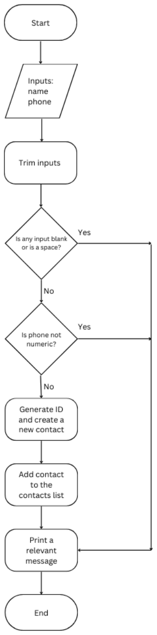
        ii. Trim inputs and check if none of the inputs are blank or are a space. If yes, print a relevant message.

        iii. Check if the phone is numeric. If not, print a relevant message.

        iv. Generate an ID and create a new dictionary with the contact information.

        v. Append the dictionary to the contacts list. Print a relevant message.

    c. Flowchart:


Start
Inputs:
name
phone
Trim inputs
Is any input blank or is a space?
Yes
No
Is phone not numeric?
Yes
No
Generate ID and create a new contact
Add contact to the contacts list
Print a relevant message
End

2. **Listing**:

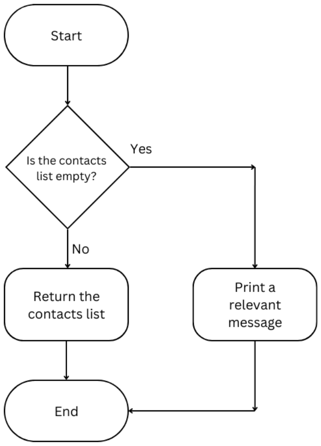
    a. Data Structure: List all contacts.

    b. Algorithm:

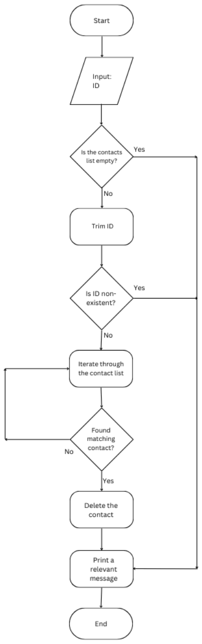
        i. If the contacts list is empty, print a relevant message.

        ii. Otherwise, print all contact details.

    c. Flowchart:



3. **Deletion**:
    a. Data Structure: Remove a contact from the list based on the ID.
    b. Algorithm:
        i. Input: ID
        ii. Check if the contacts list is empty. If it is, print a relevant message.
        iii. Trim ID and check if it is existent. If not, print a relevant message.
        iv. Iterate through the contacts list.
        v. If a contact with the matching ID is found, remove it from the list. Print a relevant message.
    c. Flowchart:



4. **Searching**:

   a. Data Structure: Search for a contact by ID, name or phone.

   b. Algorithm:

      i. Input: query

      ii. Check if the contacts list is empty. If it is, print a relevant message.

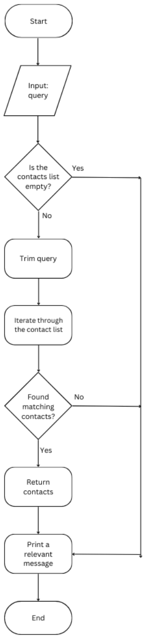
      iii. Trim query.

      iv. Iterate through the contacts list.

      v. If the query matches the contact ID or name or phone, return the contact information.

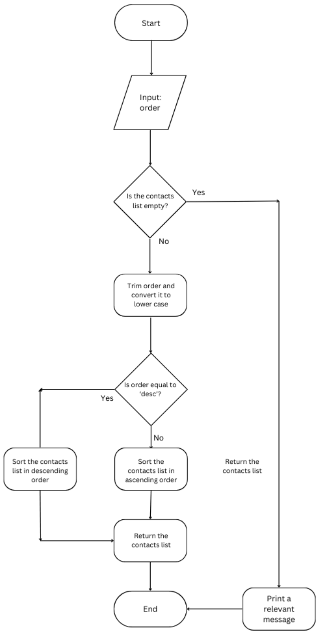
      vi. If no contacts are found, print a relevant message.

   c. Flowchart:



5. **Sorting**:
    a. Data Structure: The list of contacts will be sorted based on the contact name.
    b. Algorithm:
        i. Input: order
        ii. Check if the contacts list is empty. If it is, print a relevant message.
        iii. Trim the order and convert it to lowercase.
        iv. Check if the order is equal to ‘desc’. If it is, sort the contacts list by descending order of the contact name.
        v. Otherwise, sort the contacts list by ascending order of the contact name.
        vi. Print the sorted contacts list.

c. Flowchart:


Start
Input: order
Is the contacts list empty?
Yes
No
Trim order and convert it to lower case
Is order equal to 'desc'?
Yes
No
Sort the contacts list in descending order
Sort the contacts list in ascending order
Return the contacts list
Return the contacts list
End
Print a relevant message

Test Plan

Checkpoints

1. **Insertion**:
    a. Test adding a new contact with name and phone.
    b. Test adding a new contact with an invalid name and/or phone.
2. **Listing**:
    i. Test listing all contacts.
    ii. Test listing when there are no contacts.
3. **Deletion**:

a. Test deleting a contact by ID.

b. Test deleting a non-existent contact.

c. Test deleting a contact by ID when the contacts list is empty.

4. **Searching**:

a. Test searching for an existing contact by ID, name, and phone.

b. Test searching for a non-existent contact.

c. Test searching when the contacts list is empty.

5. **Sorting**:

a. Test sorting the contacts list by default order.

b. Test sorting the contacts list by descending order.

c. Test sorting when the contacts list is empty.

Test Cases

ID	**Description**	**Input Data**	**Expected Results**

01	Verify contact creation with a name and phone.	name=’Joey’
		phone=’9595959595’
			A new contact should be added and a relevant message should be printed.

02	Verify contact creation with a blank name and blank phone.	name=’’
		phone=’’
			A relevant message should be printed.

03	Verify contact creation with a blank name.	name=”
		phone=’8080808080’
			A relevant message should be printed.

04	Verify contact creation with a blank phone.	name=’Sam’
		phone=”
			A relevant message should be printed.

05	Verify contact creation by passing a space in name and phone.	name=’ ’
		phone=’ ’
			A relevant message should be printed.

06	Verify contact creation by passing a space in the name.	name=’ ’
		phone=’7070707070’
			A relevant message should be printed.

07	Verify contact creation by passing a space in the phone.	name=’Ron’
		phone=’ ’
			A relevant message should be printed.

08	Verify contact creation with a non-numeric phone.	name=’Jack’
		phone=’3!30303e30’
			A relevant message should be printed.

09	Verify the listing of contacts when the contacts list is not empty.	-	The contacts list should be returned.
10	Verify the listing of contacts when the contacts list is empty.	-	A relevant message should be printed.

11	Verify contact deletion with an ID.	Valid ID	A contact with a matching ID should be deleted and a relevant message should be printed.
12	Verify contact deletion with a non-existent ID.	Non-existent ID	A relevant message should be printed.
13	Verify contact deletion when the contacts list is empty.	Sample ID	A relevant message should be printed.
14	Verify the search with a query matching a single contact.	Query matching a single contact	Matching contact should be returned.
15	Verify the search with a query matching several contacts.	Query matching multiple contacts	Matching contacts should be returned.
16	Verify the search with a query matching no contacts.	Query matching no contacts	A relevant message should be printed.
17	Verify the search when the contacts list is empty.	Sample query	A relevant message should be printed.
18	Verify sorting with default order.	-	The contacts list sorted in ascending order of the name

			should be returned.
19	Verify the sorting in descending order.	order=’desc’	The contacts list sorted in descending order of the name should be returned.
20	Verify the sorting when the contacts list is empty.	order=’’	A relevant message should be printed.