**PL205 Lab Project**

**Title:** Insert strings into a Trie Data Structure

**Problem Statement:** Autocomplete using Trie data structure for a phone directory

**Team Members:**

1904, Shruti Dalvi

1929, Chetana Marathe

1930, Rochelle Martins

**Description:** In computer science, a trie, also called digital tree or prefix tree, is a kind of search tree—an ordered tree data structure used to store a dynamic set or associative array where the keys are usually strings. We are given a Trie with a set of strings stored in it. The user types in a prefix of his search query, we need to give him all recommendations to auto-complete his query based on the strings stored in the Trie.The Trie stores past searches by the users in a file.

For example if the Trie stores contact names - {“Aarya”, “Aakash”, “Alia”, “Aaron”, “Aparna”} and the User types in “Aa” then he must be shown contact card of {“Aarya”, “Aakash”, “Aaron”}

Using Trie, we can search a key in O(M) time, where M is maximum string length.

The menu consists of

- 1.1 Add a contact

The function takes the contact name and contact number from the user. The name is then stored in a trie, with the phone number stored at the end as a string after passing some necessary validation checks.

- 1.2 Modify a contact

This function is used with autocomplete suggestions for name (1.4). The user is displayed all the contacts (1.5) and is asked to enter the name of the one he/she desires to modify. If they enter an incomplete search string, the names in the trie beginning with the search string are displayed and the user is required to enter the full name based on the suggestions. Since order of adding contacts does not matter (alphabetical display) the modify reuses the delete (1.3) and insert function(1.1) [uses file handling]

- 1.3 delete a conatact

This function also uses autocomplete suggestions. The diffrence being as the function name indicates – a contact is deleted. [uses file hanling]

-1 .4 Search a contact

This function displays the contacts beginning with the user-input search string. It uses the display function (1.5) except that the name begining with the search string are displayed.

- 1.5 – Display all contacts

This function displays all the contacts in alphabetical order.

**Hardware specifications:**

Operating System: Fedora / Ubunti 18.04

**Software specifications:**

Programming Language: C

Compiler: GNU GCC Compiler

Editor: gedit/ Visual Studio Code

**Details of the Data Structures used:**

* **Trie** is a tree data structure used for storing collections of strings.

struct trie

{

struct trie \*next[CHAR\_SIZE];

int isLeaf;

char phone\_number[3][20];

};

Our trie structure has 3 members,

**struct trie \*next[CHAR\_SIZE]** which is pointer to other 55 trie structures

indices 0 - 25 are for upper case letter( A- Z), 26 – 51 are for lower case letters, 52 is for space, 53 is for apostrophe and 54 is for a full point.

**int isLeaf** when set to 1 indicates end of word, and

**char phone\_number[3][20]** holds the phone number.

* **Contact** data structure

struct contact

{

char name[25]; // holds the name of the contact

char phone\_num[12]; // holds the phone number

};

**TEST CASES**

* **ADD CONTACT FUNCTION**

Enter name:

|  |  |  |  |
| --- | --- | --- | --- |
| #TC | TC VALUE | TC DESCRIPTION | TC RESULT |
| 1 | Ma | Length of name should be at least 3 characters. | INVALID |
| 2 | Maria2 / Maria- | Name should not contain any numbers or special chracters other than an apostrophe(') and a full point(.). | INVALID |
| 3.1 | Maria | Length of the Name is greater than 3 and has no numbers or special chracters(other than an apostrophe(') and a full point(.)). | VALID |
| 3.2 | Maria | Name should not be identical. | INVALID |

Enter phone number:

|  |  |  |  |
| --- | --- | --- | --- |
| #TC | TC VALUE | TC DESCRIPTION | TC RESULT |
| 1 | 9843 / 98765432100 | Length of phone number sholud not be less than or greater than 10. It should have exactly 10 digits. | INVALID |
| 2 | 984356789w | Phone number should contain only numbers and not characters. | INVALID |
| 3.1 | 9876543210 | Phone number has no charaters and length is equal to 10. Also phone number is unique | VALID |
| 3.2 | 9876543210 | Phone numbers should not be identical. | INVALID |

* **MODIFY FUNCTION**

|  |  |
| --- | --- |
| **CONTACTS** | |
| **NAME** | **PHONE NUMBER** |
| Rihanna | 9090190902 |
| Riya | 7878678785 |

MODIFY BY NUMBER

Enter contact number you want to modify:-

|  |  |  |  |
| --- | --- | --- | --- |
| #TC | TC VALUE | TC DESCRIPTION | TC RESULT |
| 1 | 9090909090 | Contact number isnt stored in list.Contact with this name does not exist. | INVALID |
| 2 | 9090190902 | Contact number is present in the list. | VALID |

MODIFY BY NAME

Enter contact name you want to modify:-

|  |  |  |  |
| --- | --- | --- | --- |
| #TC | TC VALUE | TC DESCRIPTION | TC RESULT |
| 1 | siya | Contact name isnt stored in list.Contact with this name does not exist. | INVALID |
| 2 | Rihanna | Contact name is present in the list. | VALID |

* **DELETE CONTACT FUNCTION**

|  |  |
| --- | --- |
| **CONTACTS** | |
| **NAME** | **PHONE NUMBER** |
| Rihanna | 9090190902 |
| Riya | 7878678785 |

DELETE BY NUMBER

Enter contact number you want to delete:-

|  |  |  |  |
| --- | --- | --- | --- |
| #TC | TC VALUE | TC DESCRIPTION | TC RESULT |
| 1 | 9090909090 | Contact number isnt stored in list. | INVALID |
| 2 | 9090190902 | Contact number is present in the list. | VALID |

DELETE BY NAME

Enter contact nane you want to delete:-

|  |  |  |  |
| --- | --- | --- | --- |
| #TC | TC VALUE | TC DESCRIPTION | TC RESULT |
| 1 | siya | Contact name isnt stored in list. | INVALID |
| 2 | Rihanna | Contact name is present in the list. | VALID |

* **SEARCH FUNCTION**

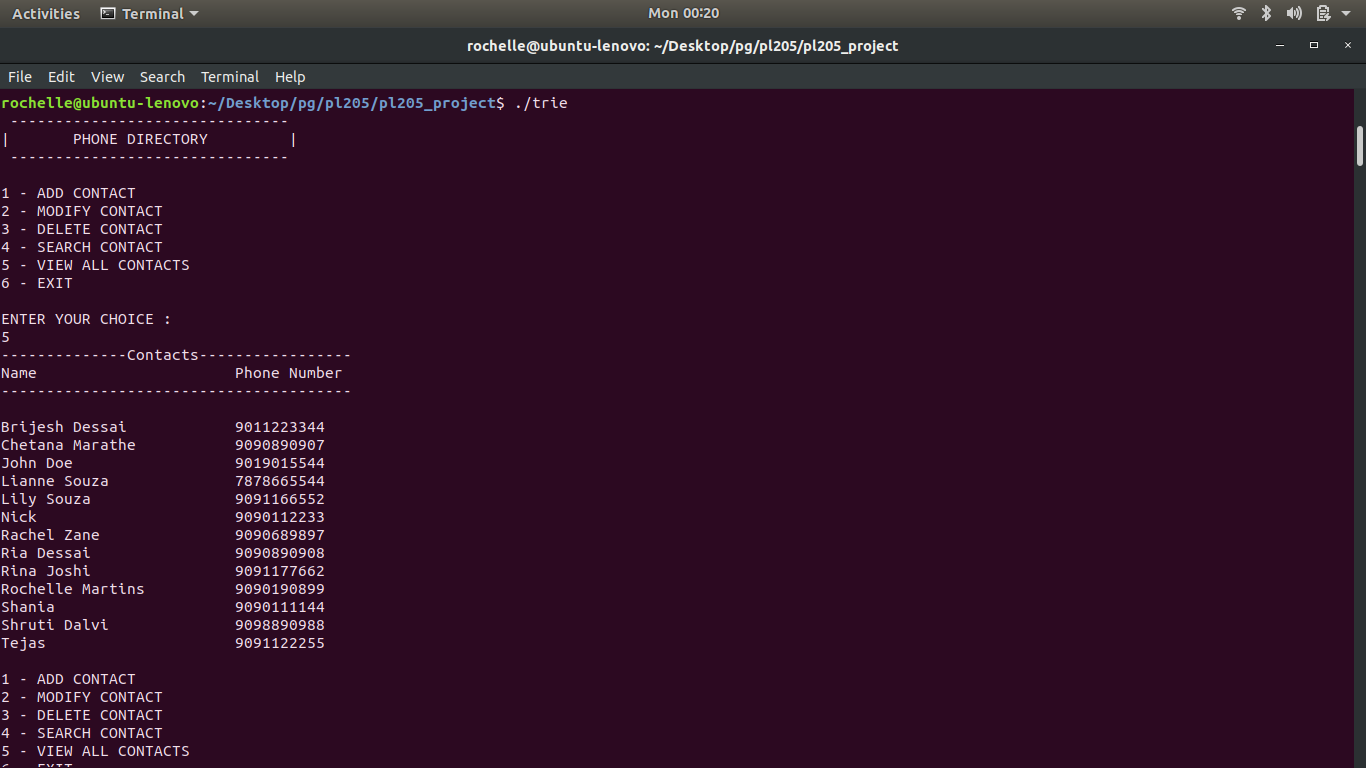
|  |  |
| --- | --- |
| **CONTACTS** | |
| **NAME** | **PHONE NUMBER** |
| Rihanna | 9090190902 |
| Riya | 7878678785 |

Enter name of the contact you want to search:-

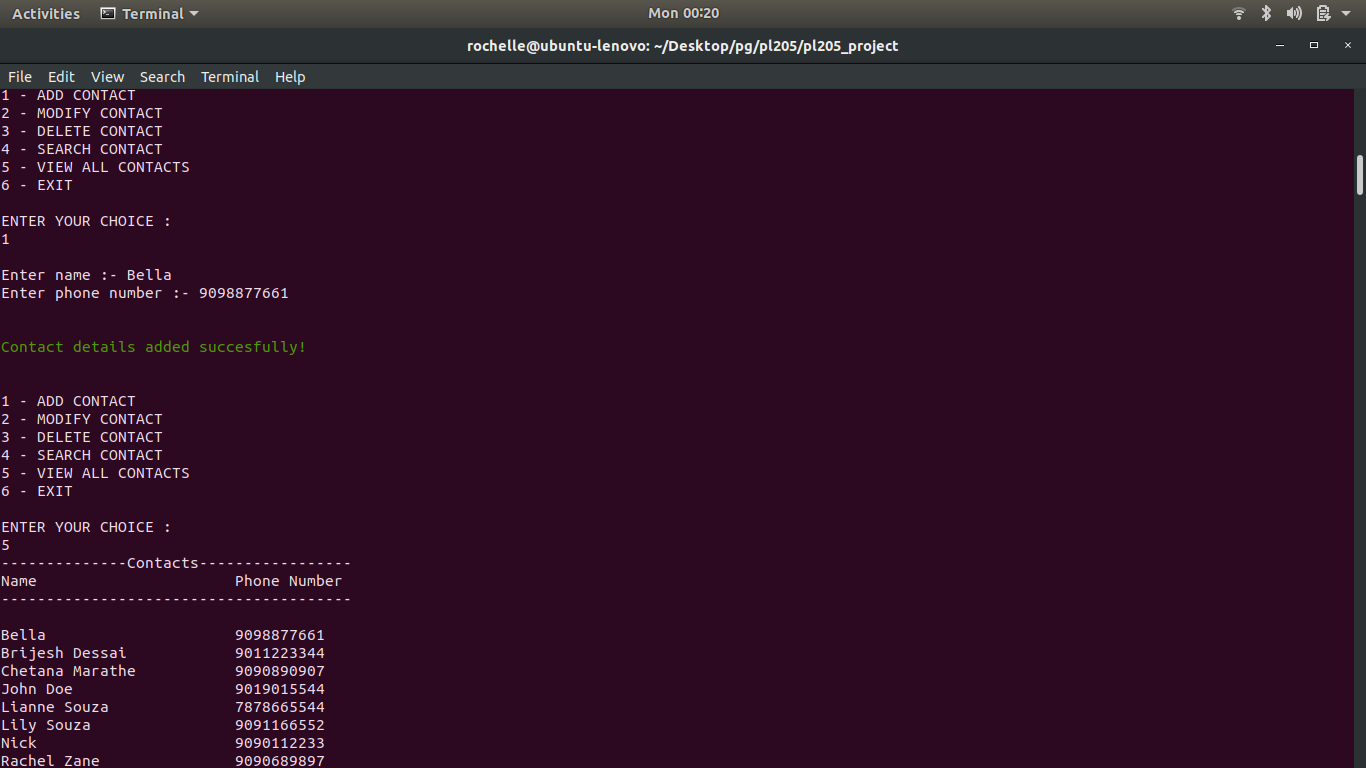
|  |  |  |  |
| --- | --- | --- | --- |
| #TC | TC VALUE | TC DESCRIPTION | TC RESULT |
| 1 | RIYA / riya | Search stings does not matches with the stored string. | INVALID |
| 2 | Riya | Search string matches. | VALID |
| 3 | R | All the strings that starts with ‘R’ will b displayed. | VALID |

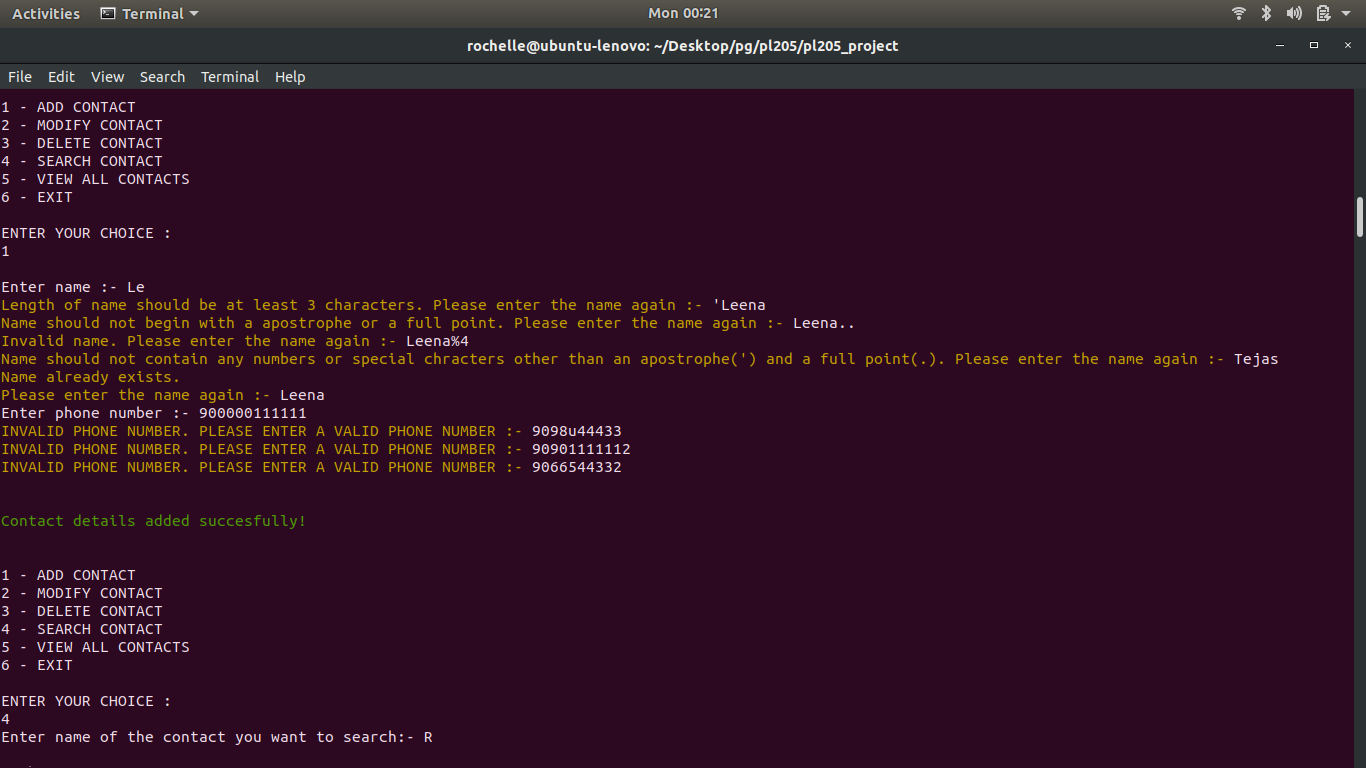
**Screenshots:**

**Option 5. View All Contacts**

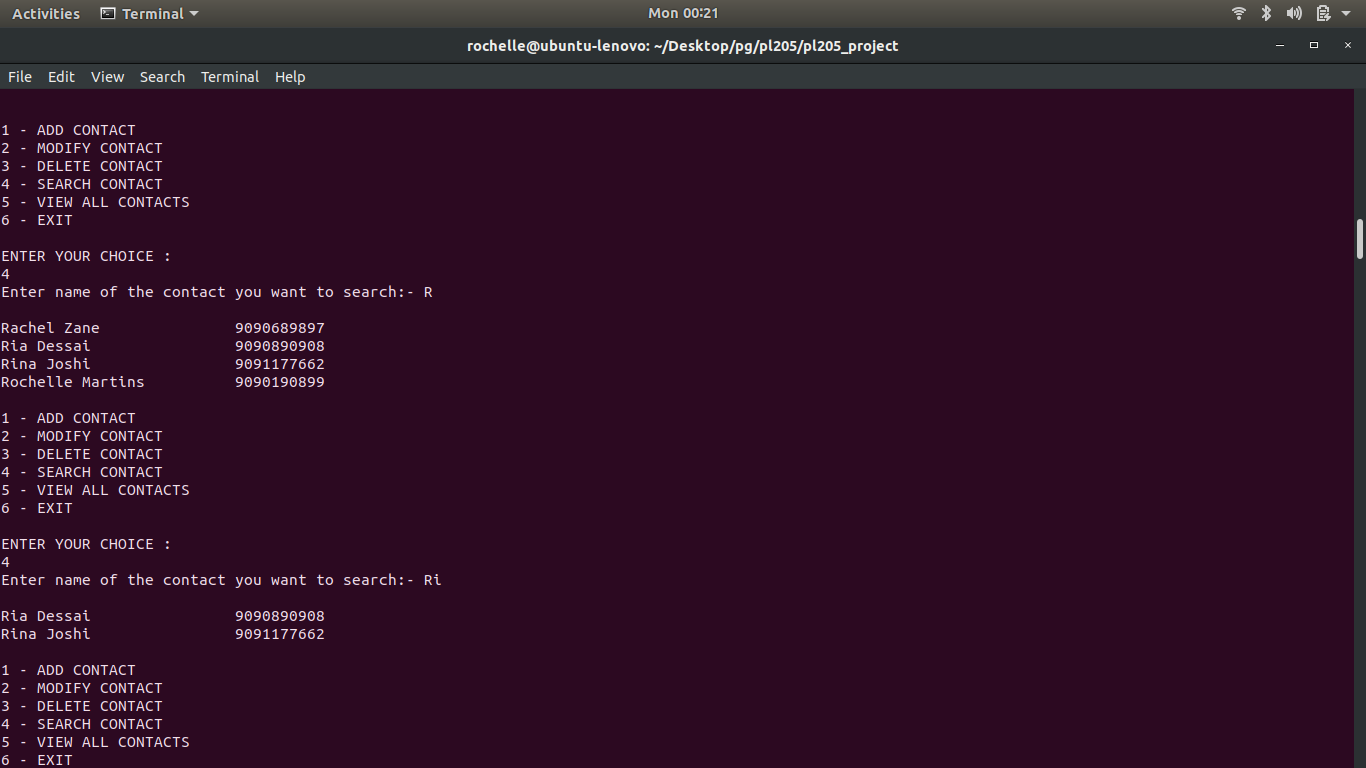


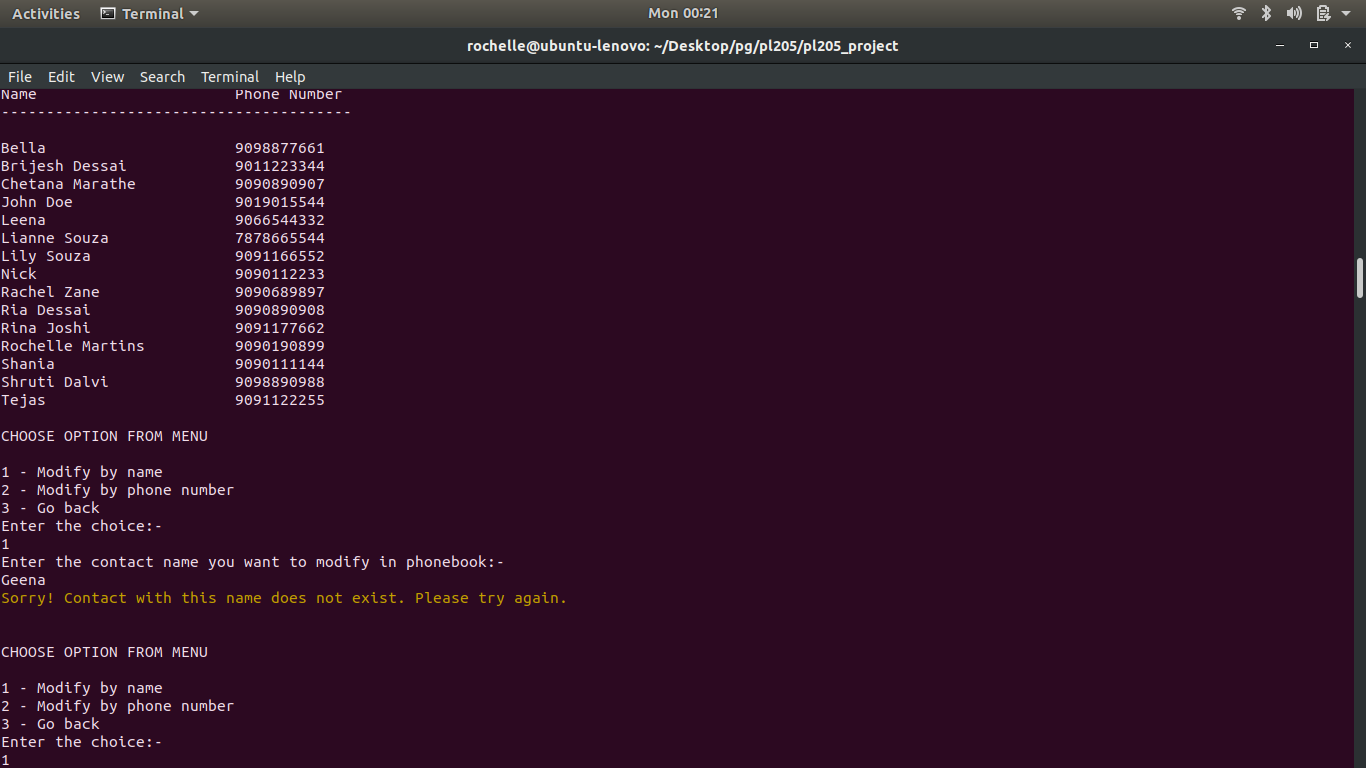
**Option 1: Add A Contact**

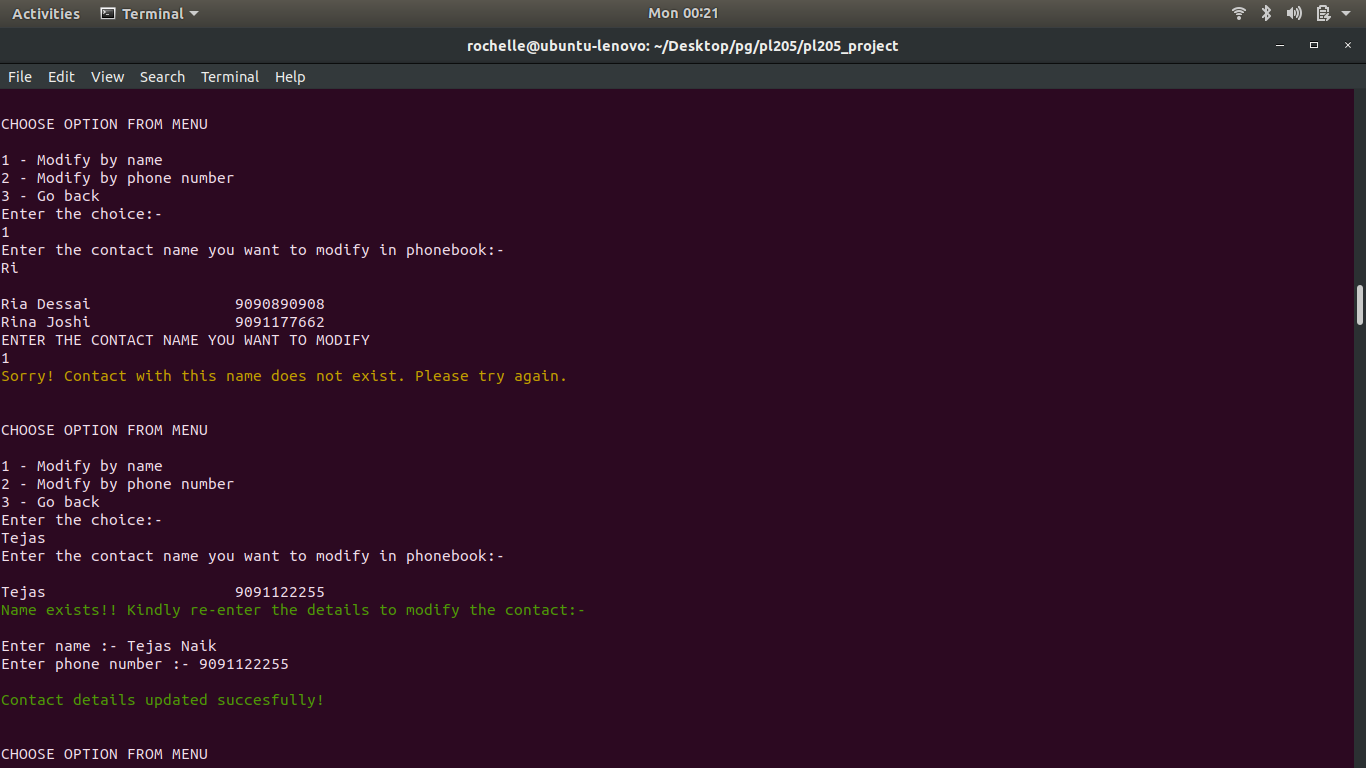


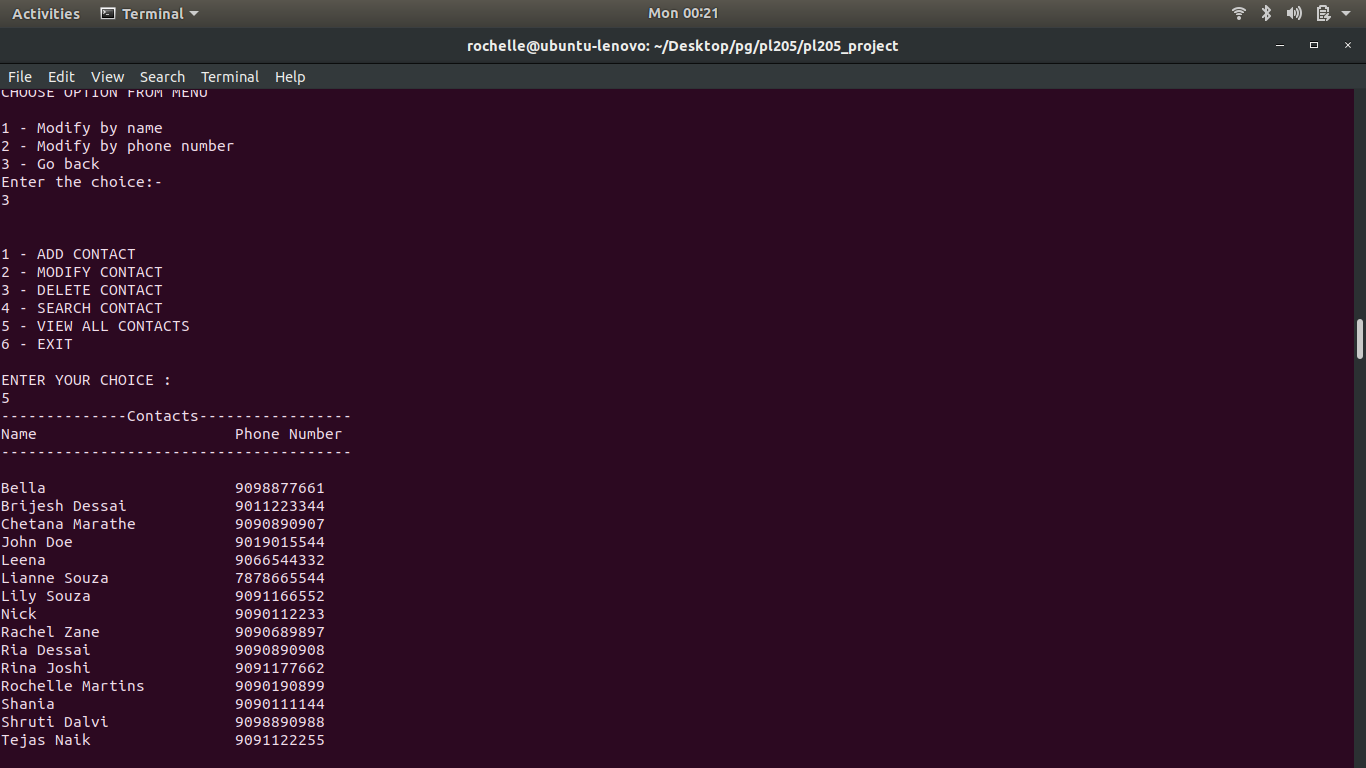
**Validations**

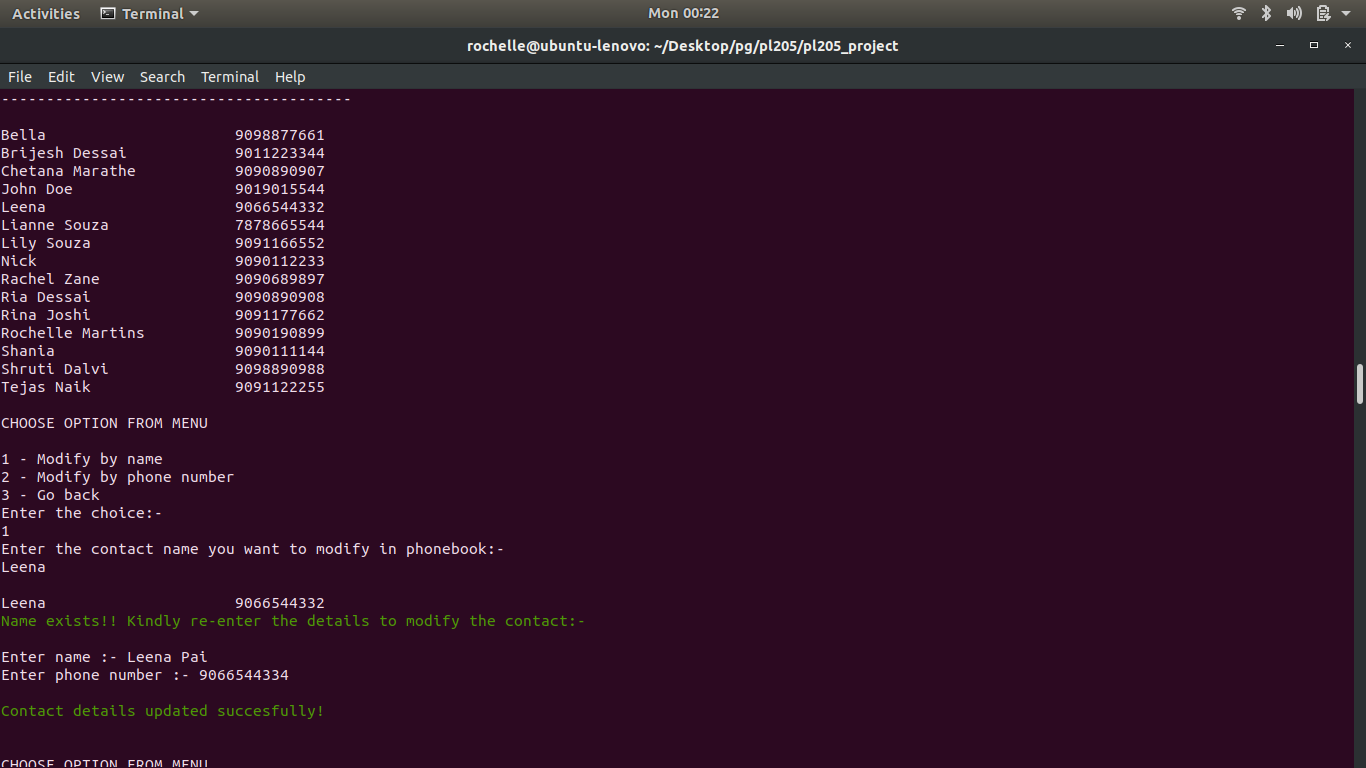
**Option 4: Search**

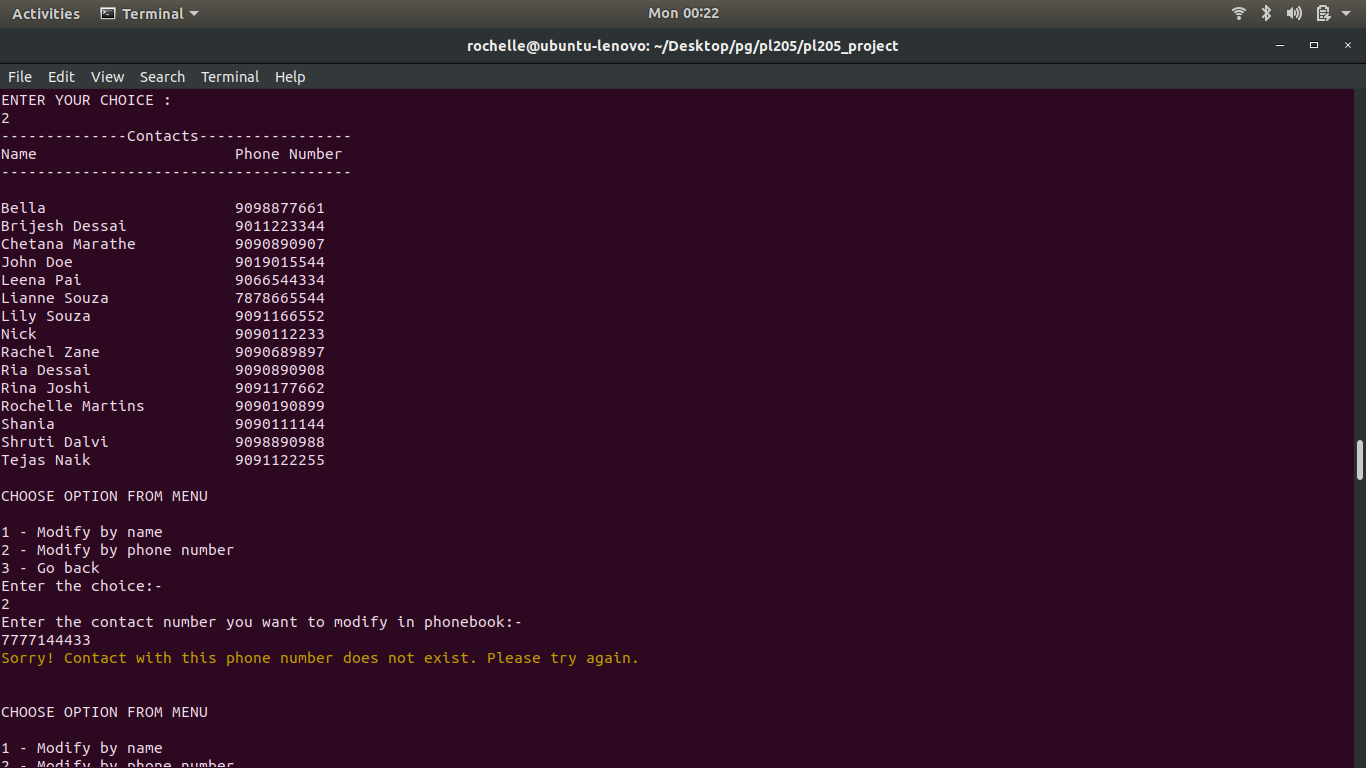


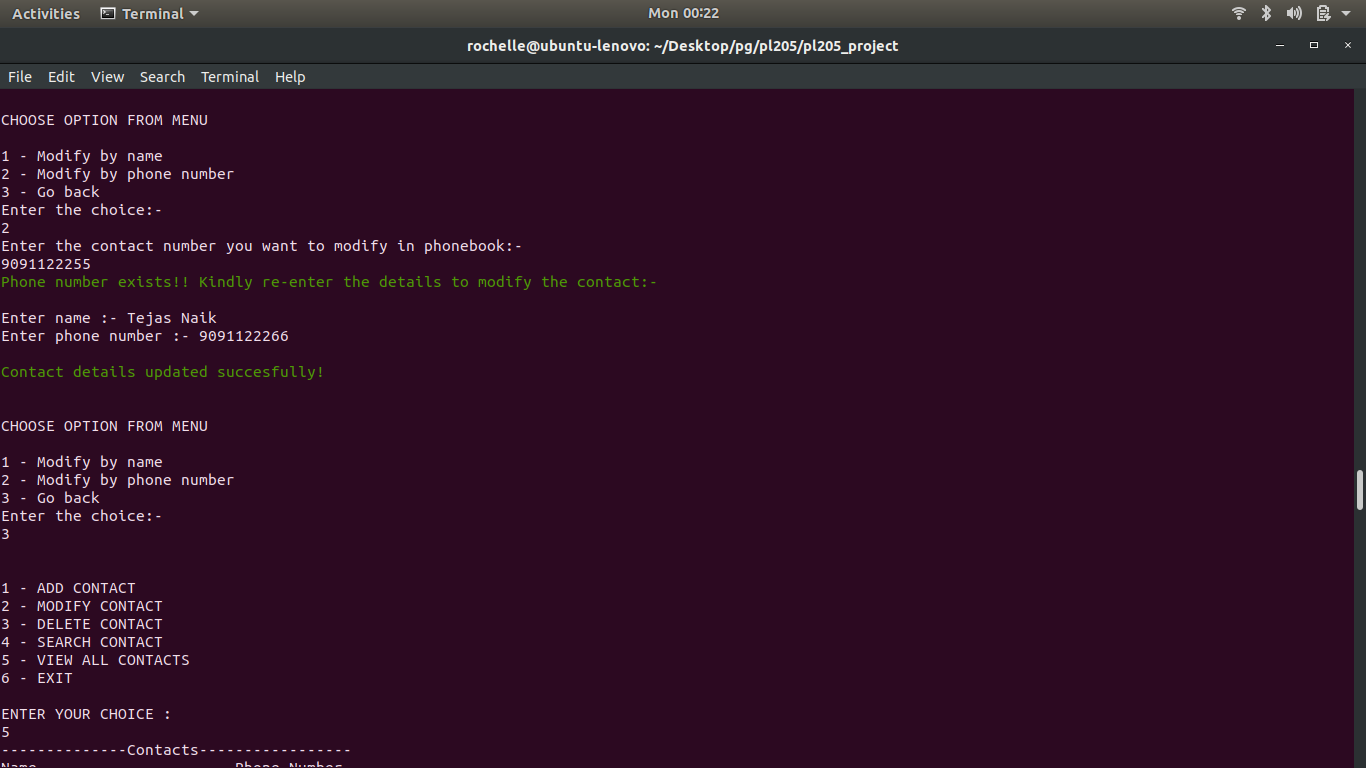
**Option 5: Modify Contact**



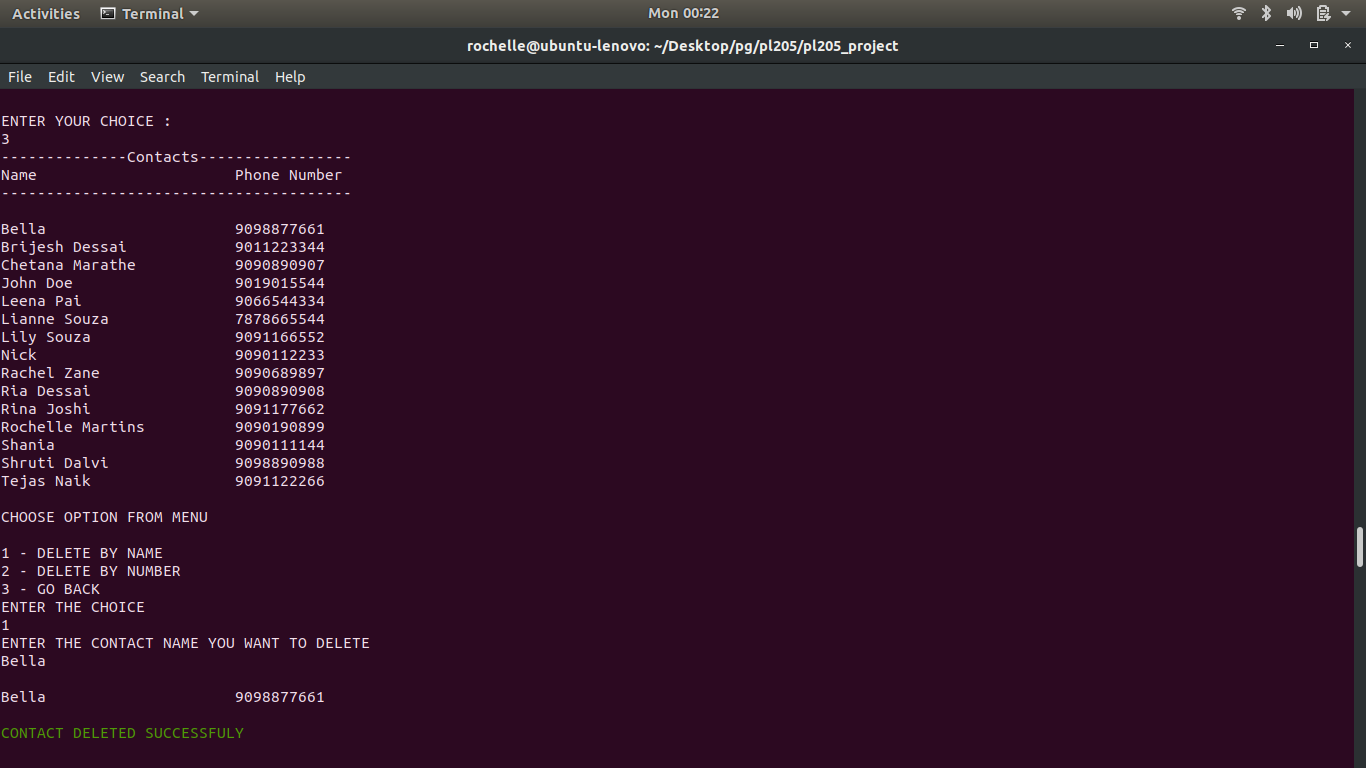


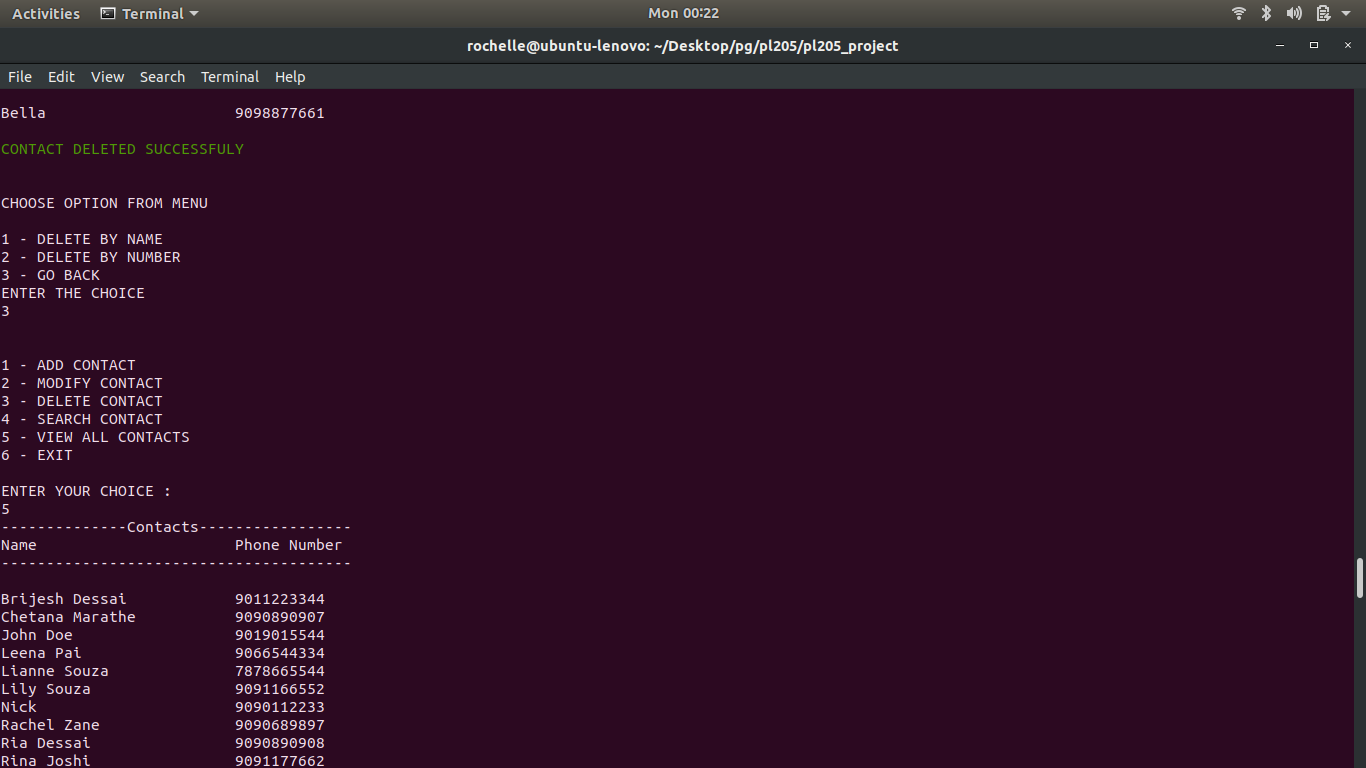


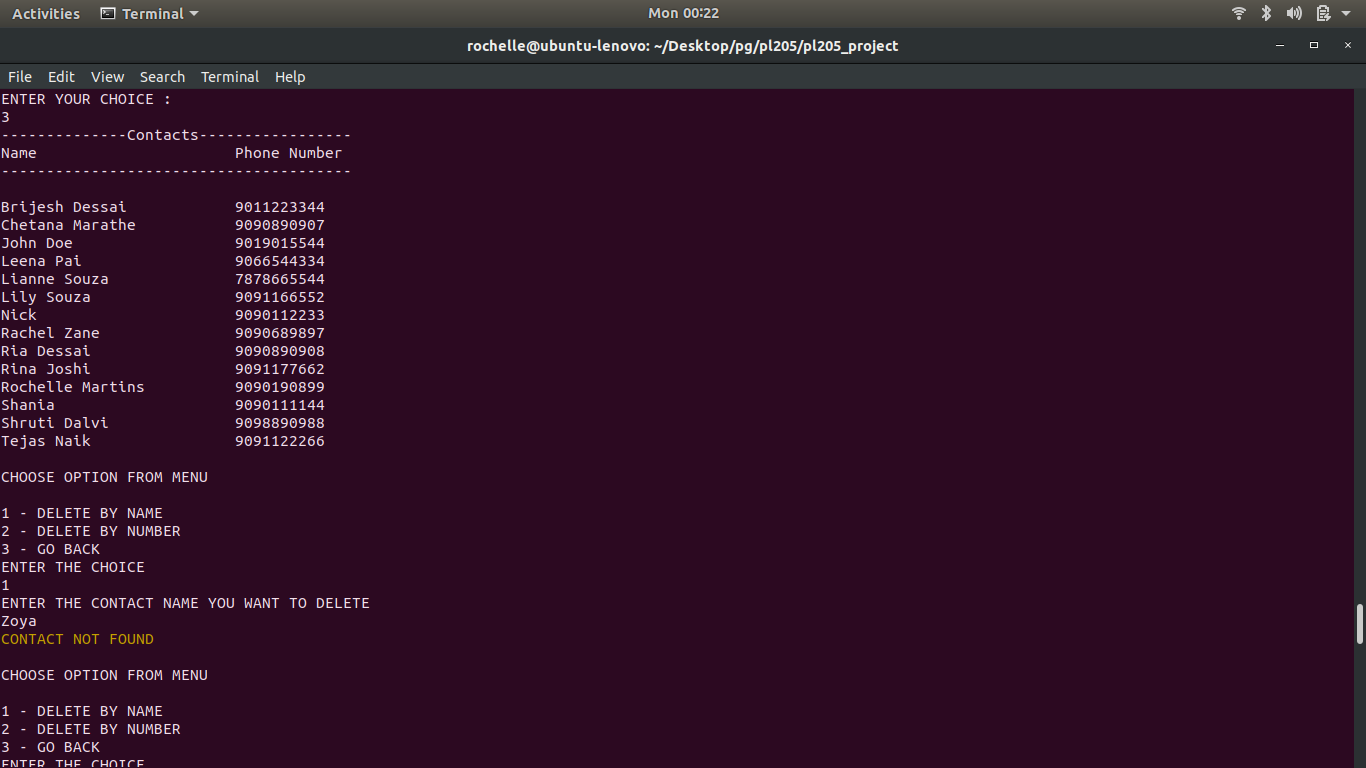


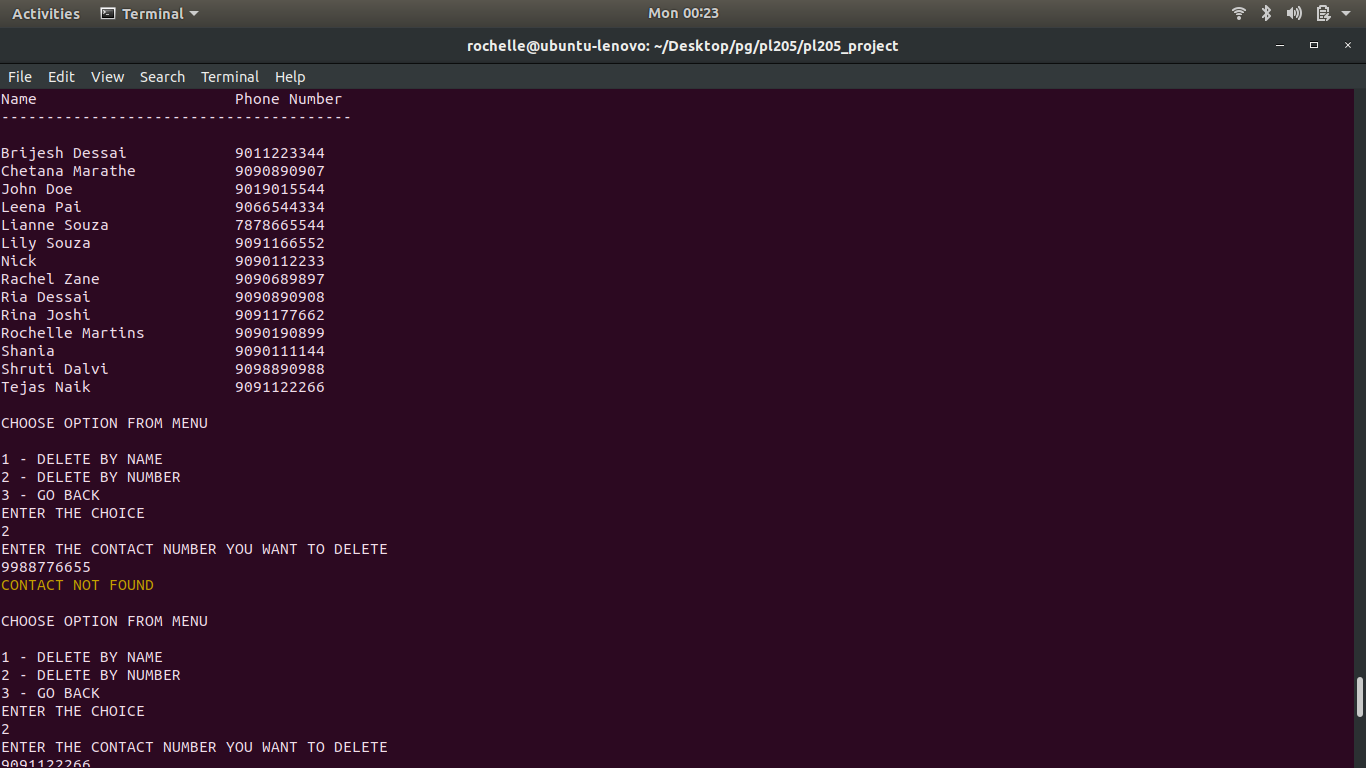


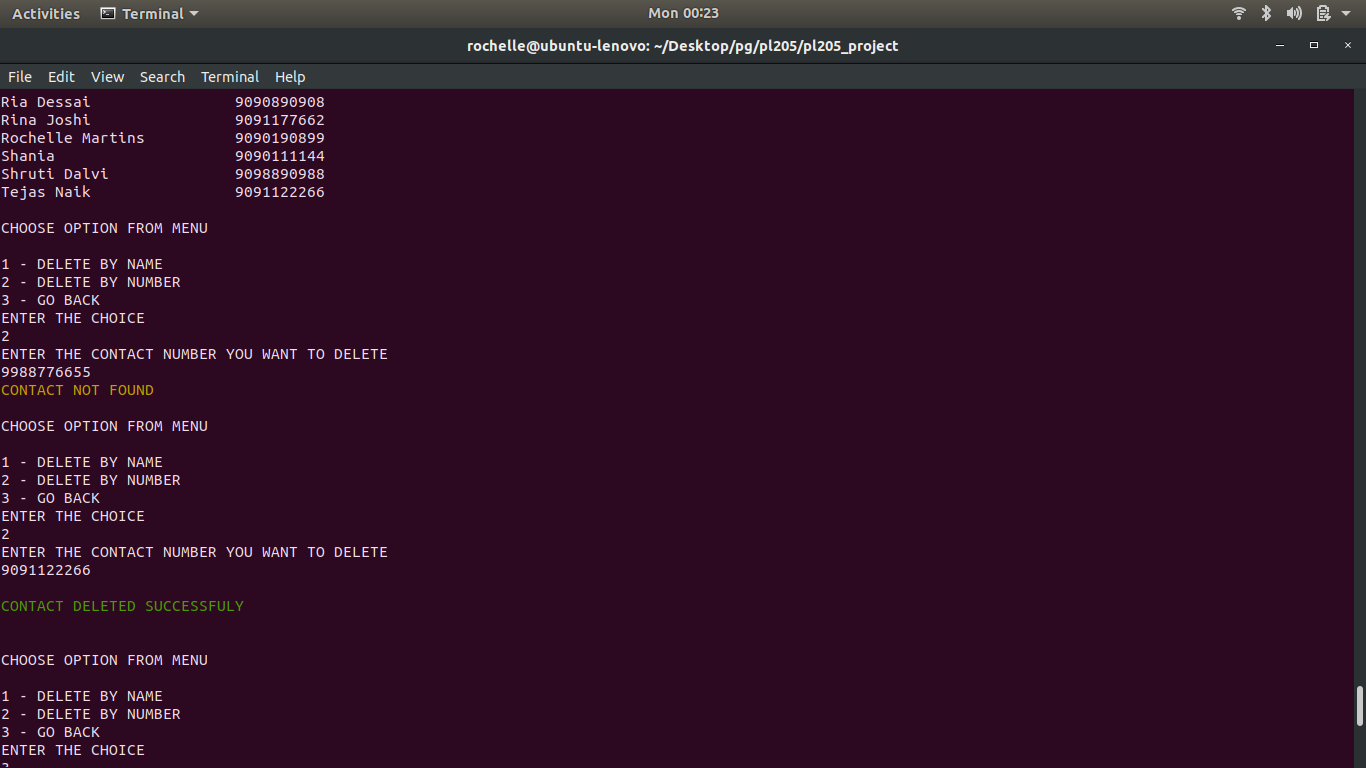
**Option 3: Delete function**

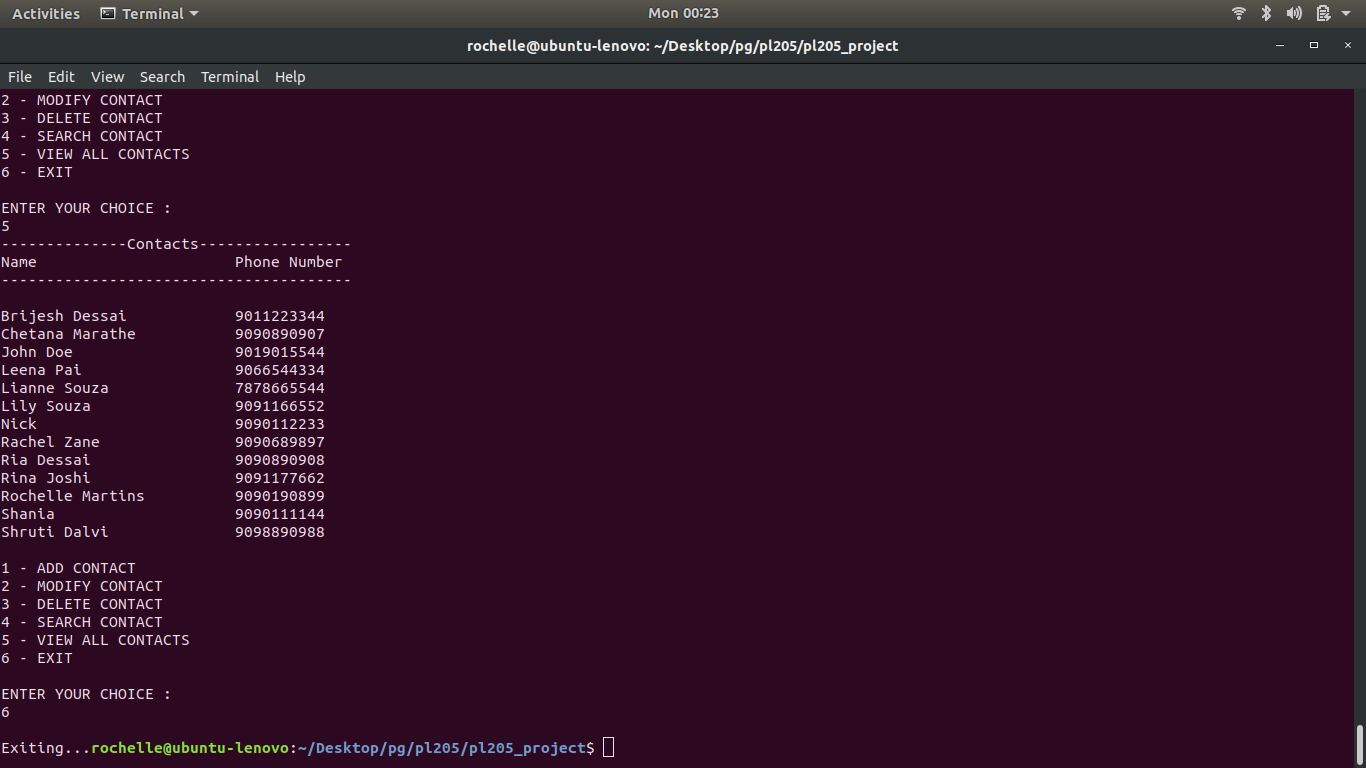












**Division of work:**

Code

Functions:

menu – Chetana, Shruti

add a contact, insert to trie – Rochelle

modify – Chetana, Rochelle (+ autocomplete, minor improvements)

delete and related fuctions – Shruti, Rochelle (+autocomplete, minor improvements)

write to file – Rochelle, Shruti

read from file – Rochelle, Chetana

validation for name – Chetana, Rochelle (case 2, 4), Shruti (case 5)

validation for number – Shruti

check duplicate name and number – Shruti

search (autocomplete), display - Rochelle

Data Structures

Rochelle

Enhancements

Color codes for messages – Chetana

Validation macros – Rochelle

Test Cases

Shruti – (written and tested)

Testing

- Chetana, Shruti, Rochelle

**Problems faced and their solution**

The maindisadvantage of **tries** is that they need a lot of memory for storing the strings. For each node we have too many node pointers(equal to number of characters of the alphabet).

Nevertheless, it was very interesting to learn about this data structure.

So that the user can save the contact however they please ( in upper case as well as lower case, with apostrophes and abbreviated names) we introduced a 55 member array instead of a 26-member array normally used.

**Scope of Improvment**

- Store and search the phone number also as a string

- Use a 26-member array and store the name without regard to case (like a dictionary)

- delete and modify using the trie ( algorithm ready, execution pending)

- store multiple numbers per contact