Introduction

A singly linked list is a **type of linked list that is unidirectional**, that is, it can be traversed in only one direction from head to the last node (tail).A single node contains data and a pointer to the next node which helps in maintaining the structure of the list.

You can also decrease and increase the linked list at run-time. That is, you can allocate and deallocate memory at run-time itself. In this, you can easily do insertion and deletion functions. That is, you can easily insert and delete the node.

Background

We have basically given 5 options to the user-

1. Create a new entry.

2. Delete an entry.

3. Find an entry and show its information.

4. Save the whole phonebook in a file.

5. Exit

These functionalities helps the user to easily enter the details, edit it, add new contacts, delete, display and save phonebook in a file without any difficulty. We have used C++ language in this project and used these concepts mainly in out project.

1. Class

2. Functions

3. Strings

4. Pointers

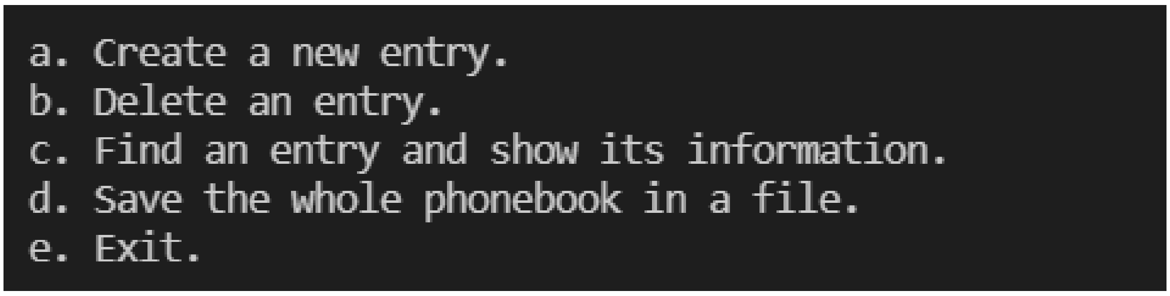
5. Doubly linked list

Design

We first made a basic layout and then added functionalities to it as and when needed according to the level of complexity we needed for a group project and upto the standards of our esteemed college. We have included many other functions like create, delete, show, save in the file to make our program more efficient.

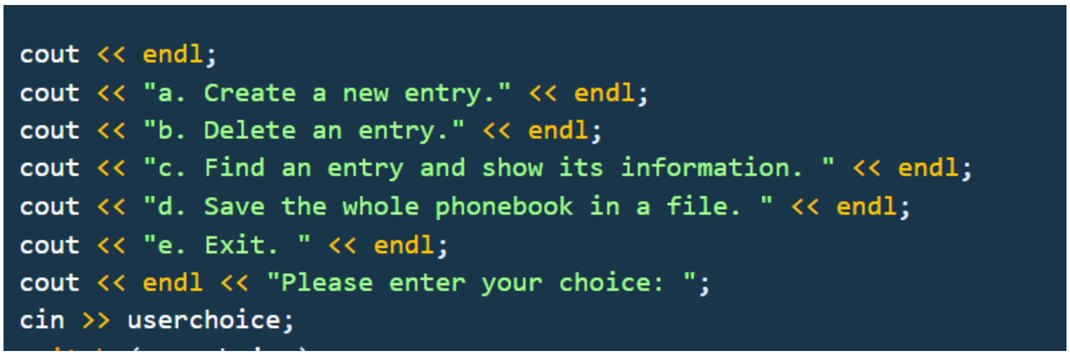
Implementation

Creating the Phonebook



```
a. Create a new entry.
b. Delete an entry.
c. Find an entry and show its information.
d. Save the whole phonebook in a file.
e. Exit.
```

Code:-

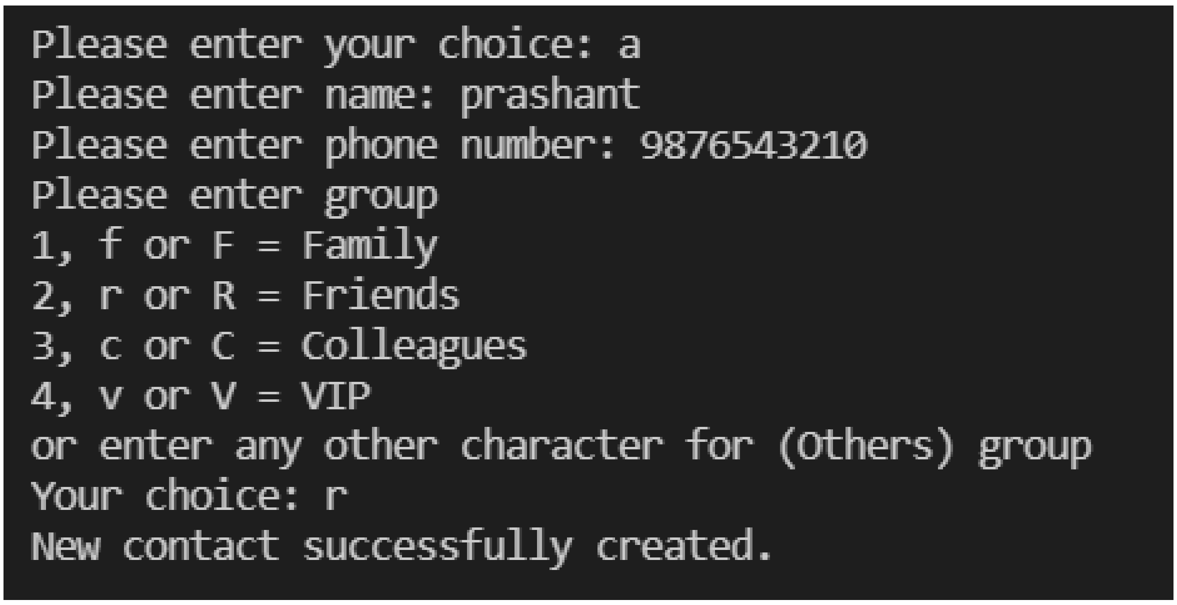


```
cout << endl;
cout << "a. Create a new entry." << endl;
cout << "b. Delete an entry." << endl;
cout << "c. Find an entry and show its information. " << endl;
cout << "d. Save the whole phonebook in a file. " << endl;
cout << "e. Exit. " << endl;
cout << endl << "Please enter your choice: ";
cin >> userchoice;
```

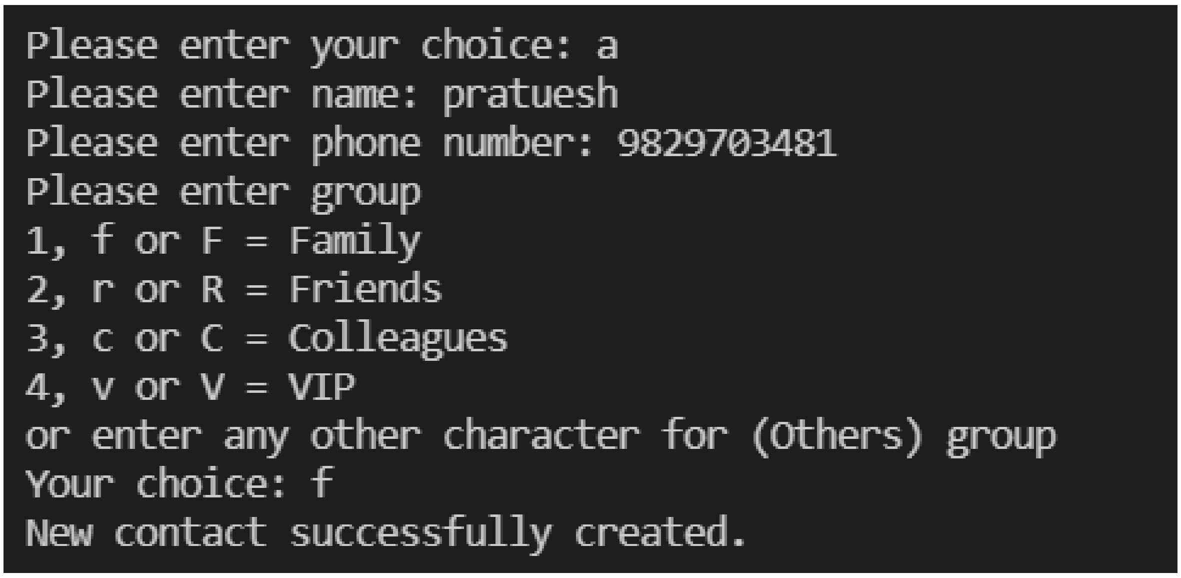
Firstly user is asked to choose one of the five options given to them. User can create a entry, Delete an entry, Find an entry and show its information etc.

After entering desired number of entries user can delete any entry. First program search for the specified name and if present in the phonebook it deletes it. We used file handling since we don’t want all the data to go away even after the compilation is done, so we created a .txt file and stored all the nodes in it so that the “contact list” can be readily accessed anytime.

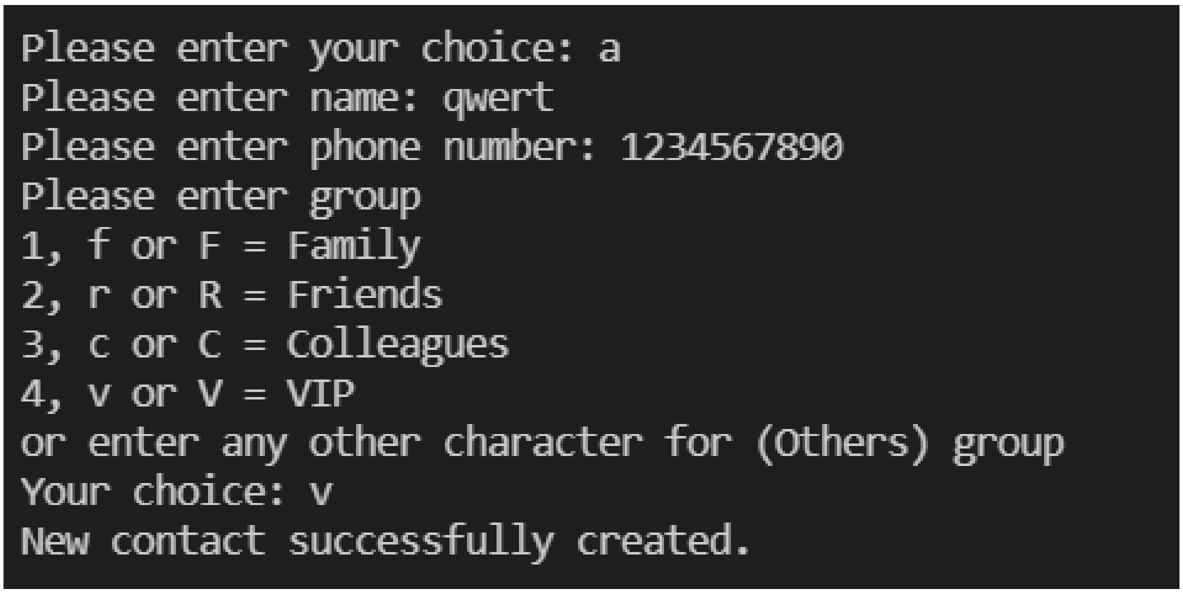
Creating entries in the phonebook



```
Please enter your choice: a
Please enter name: prashant
Please enter phone number: 9876543210
Please enter group
1, f or F = Family
2, r or R = Friends
3, c or C = Colleagues
4, v or V = VIP
or enter any other character for (Others) group
Your choice: r
New contact successfully created.
```

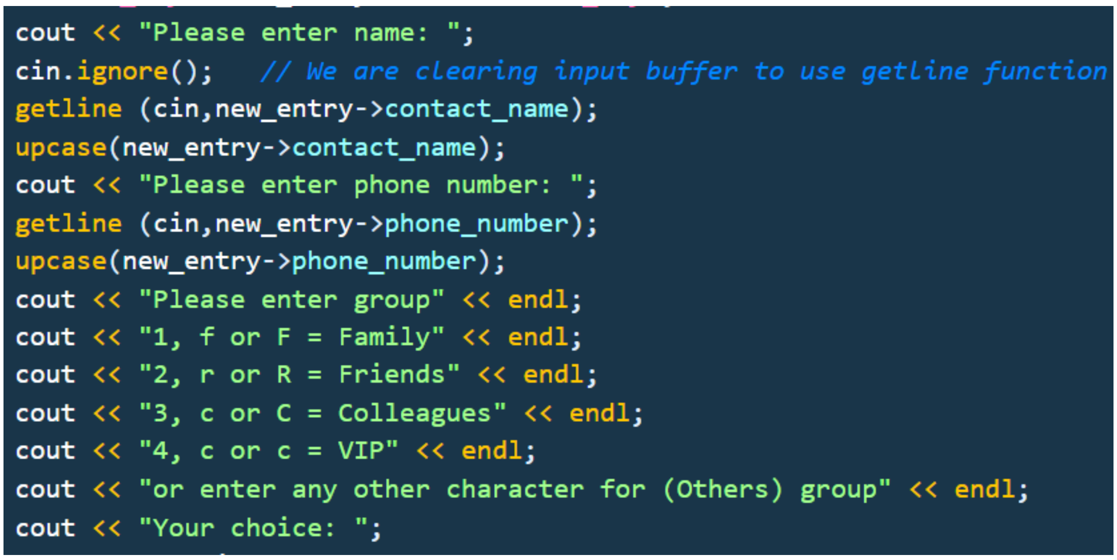


```
Please enter your choice: a
Please enter name: pratuesh
Please enter phone number: 9829703481
Please enter group
1, f or F = Family
2, r or R = Friends
3, c or C = Colleagues
4, v or V = VIP
or enter any other character for (Others) group
Your choice: f
New contact successfully created.
```



```
Please enter your choice: a
Please enter name: qwert
Please enter phone number: 1234567890
Please enter group
1, f or F = Family
2, r or R = Friends
3, c or C = Colleagues
4, v or V = VIP
or enter any other character for (Others) group
Your choice: v
New contact successfully created.
```

Code:-

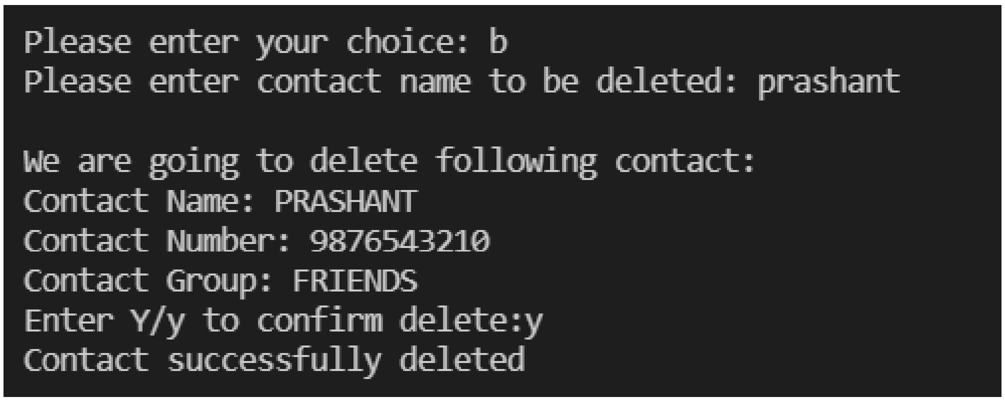


```cpp
cout << "Please enter name: ";
cin.ignore();   // We are clearing input buffer to use getline function
getline (cin,new_entry->contact_name);
upcase(new_entry->contact_name);
cout << "Please enter phone number: ";
getline (cin,new_entry->phone_number);
upcase(new_entry->phone_number);
cout << "Please enter group" << endl;
cout << "1, f or F = Family" << endl;
cout << "2, r or R = Friends" << endl;
cout << "3, c or C = Colleagues" << endl;
cout << "4, c or c = VIP" << endl;
cout << "or enter any other character for (Others) group" << endl;
cout << "Your choice: ";
```

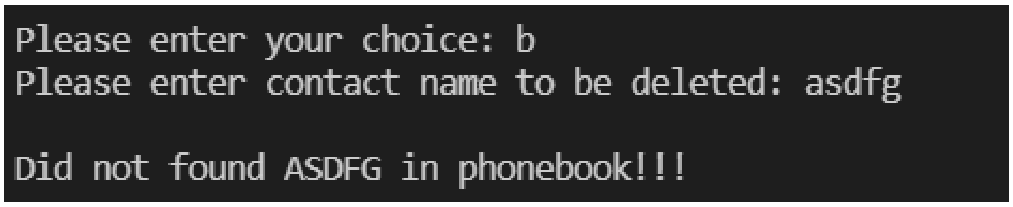
User is asked to enter their name, correct phone number and then they are asked to in which group they want their contact to be shared(Family, Friends, Colleagues, VIP or others). After entering “New contact successfully created “ message is shown.

Delete an entry

User is asked to enter the name of the contact to be deleted. If the name is already present in the phonebook then the program displays all the details of the contact and asks user to confirm deletion. If the name is not present in the phonebook it displays "Did not found name in the phonebook".

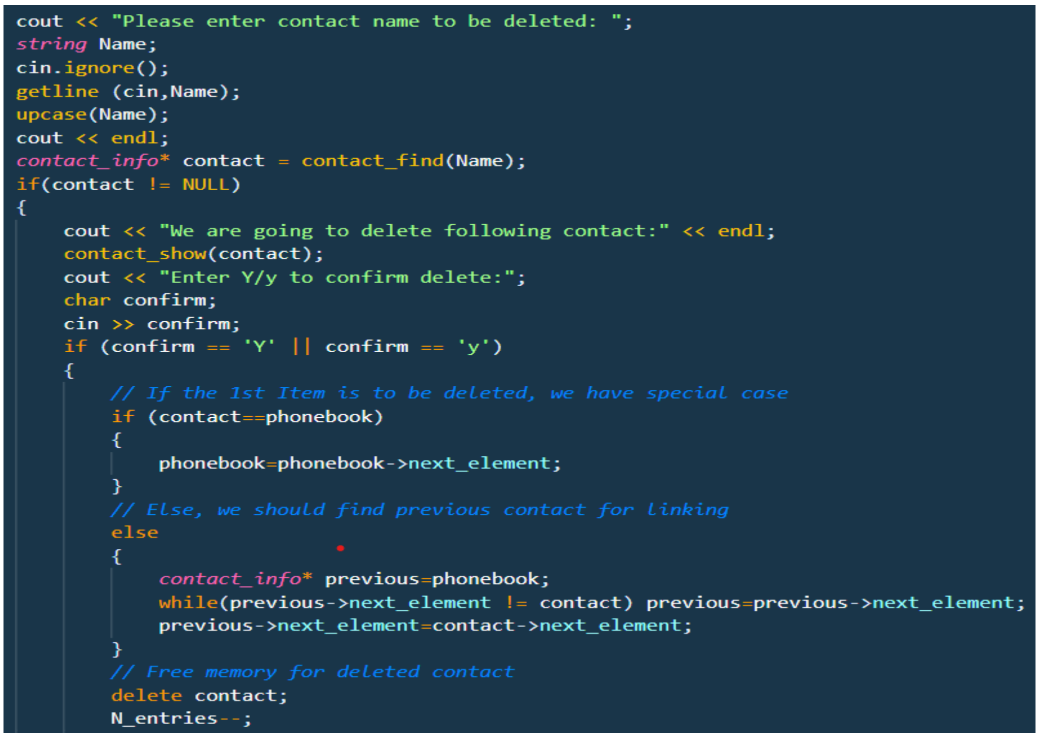


```
Please enter your choice: b
Please enter contact name to be deleted: prashant

We are going to delete following contact:
Contact Name: PRASHANT
Contact Number: 9876543210
Contact Group: FRIENDS
Enter Y/y to confirm delete:y
Contact successfully deleted
```



```
Please enter your choice: b
Please enter contact name to be deleted: asdfg

Did not found ASDFG in phonebook!!!
```

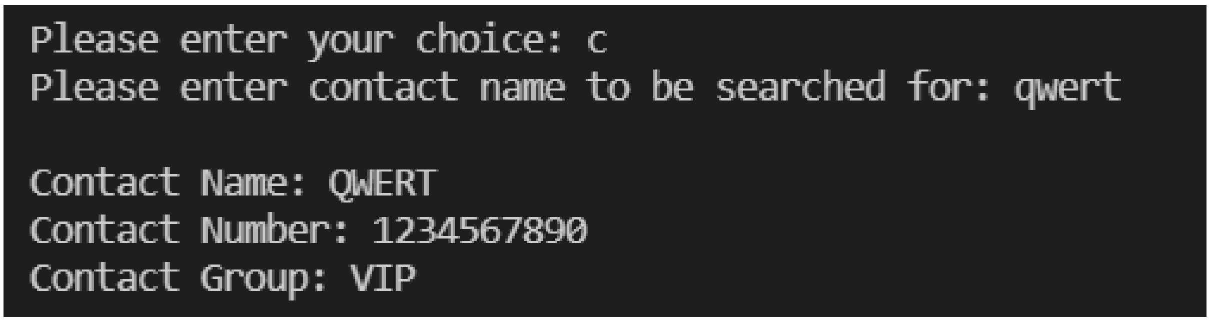
Code:-



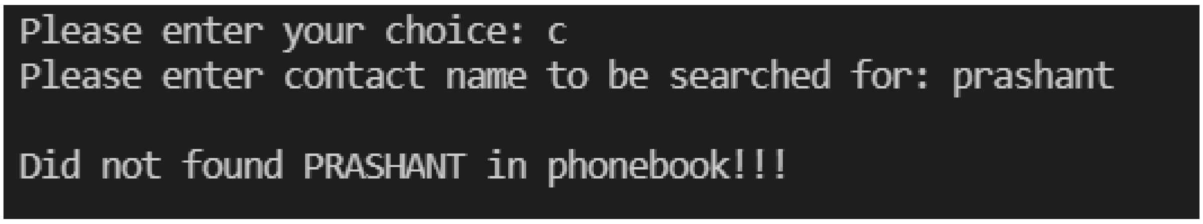
```
cout << "Please enter contact name to be deleted: ";
string Name;
cin.ignore();
getline (cin,Name);
upcase(Name);
cout << endl;
contact_info* contact = contact_find(Name);
if(contact != NULL)
{
    cout << "We are going to delete following contact:" << endl;
    contact_show(contact);
    cout << "Enter Y/y to confirm delete:";
    char confirm;
    cin >> confirm;
    if (confirm == 'Y' || confirm == 'y')
    {
        // If the 1st Item is to be deleted, we have special case
        if (contact==phonebook)
        {
            phonebook=phonebook->next_element;
        }
        // Else, we should find previous contact for linking
        else
        {
            contact_info* previous=phonebook;
            while(previous->next_element != contact) previous=previous->next_element;
            previous->next_element=contact->next_element;
        }
        // Free memory for deleted contact
        delete contact;
        N_entries--;
```

Find an entry and show its information

This functions asks to user to enter the name of the contact he/she wants to search find information about. If the entered name is not present in the phonebook it displays “Did not found name in phonebook!!”. If the name is present in the phonebook the program displays name contact number and the group of the given contact.

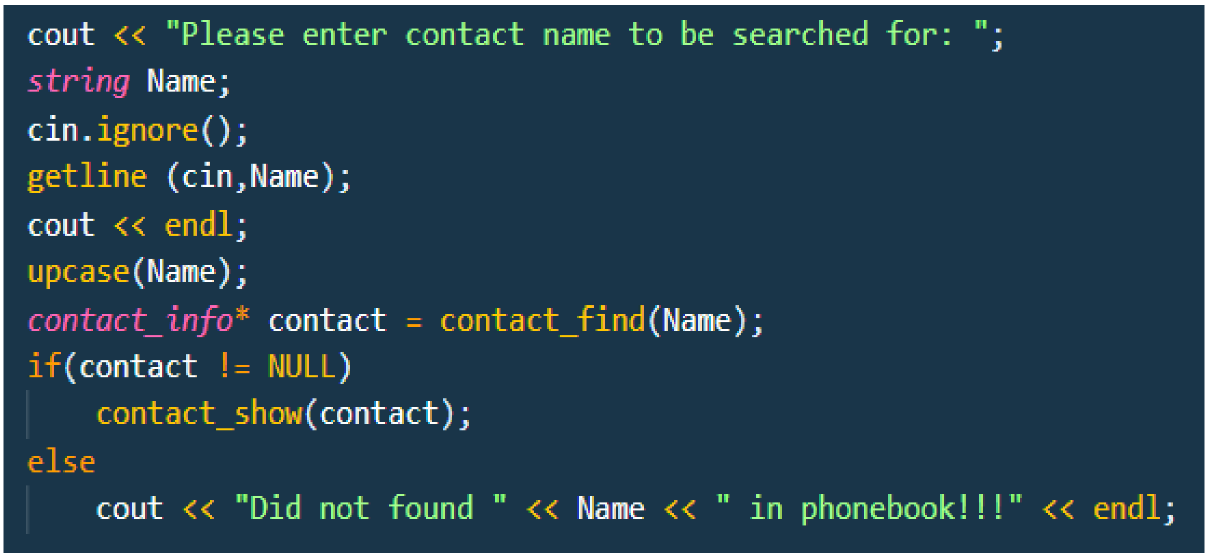


```
Please enter your choice: c
Please enter contact name to be searched for: qwert

Contact Name: QWERT
Contact Number: 1234567890
Contact Group: VIP
```



```
Please enter your choice: c
Please enter contact name to be searched for: prashant

Did not found PRASHANT in phonebook!!!
```

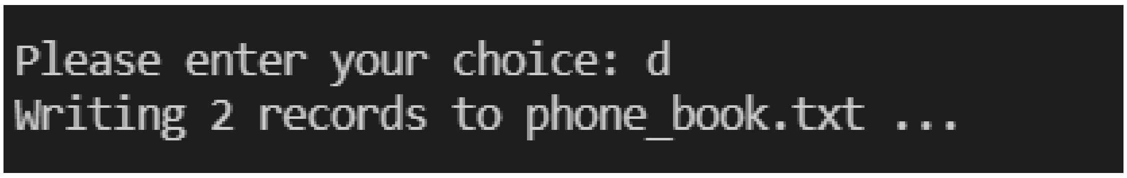
Code:-



```cpp
cout << "Please enter contact name to be searched for: ";
string Name;
cin.ignore();
getline (cin,Name);
cout << endl;
upcase(Name);
contact_info* contact = contact_find(Name);
if(contact != NULL)
    contact_show(contact);
else
    cout << "Did not found " << Name << " in phonebook!!!" << endl;
```

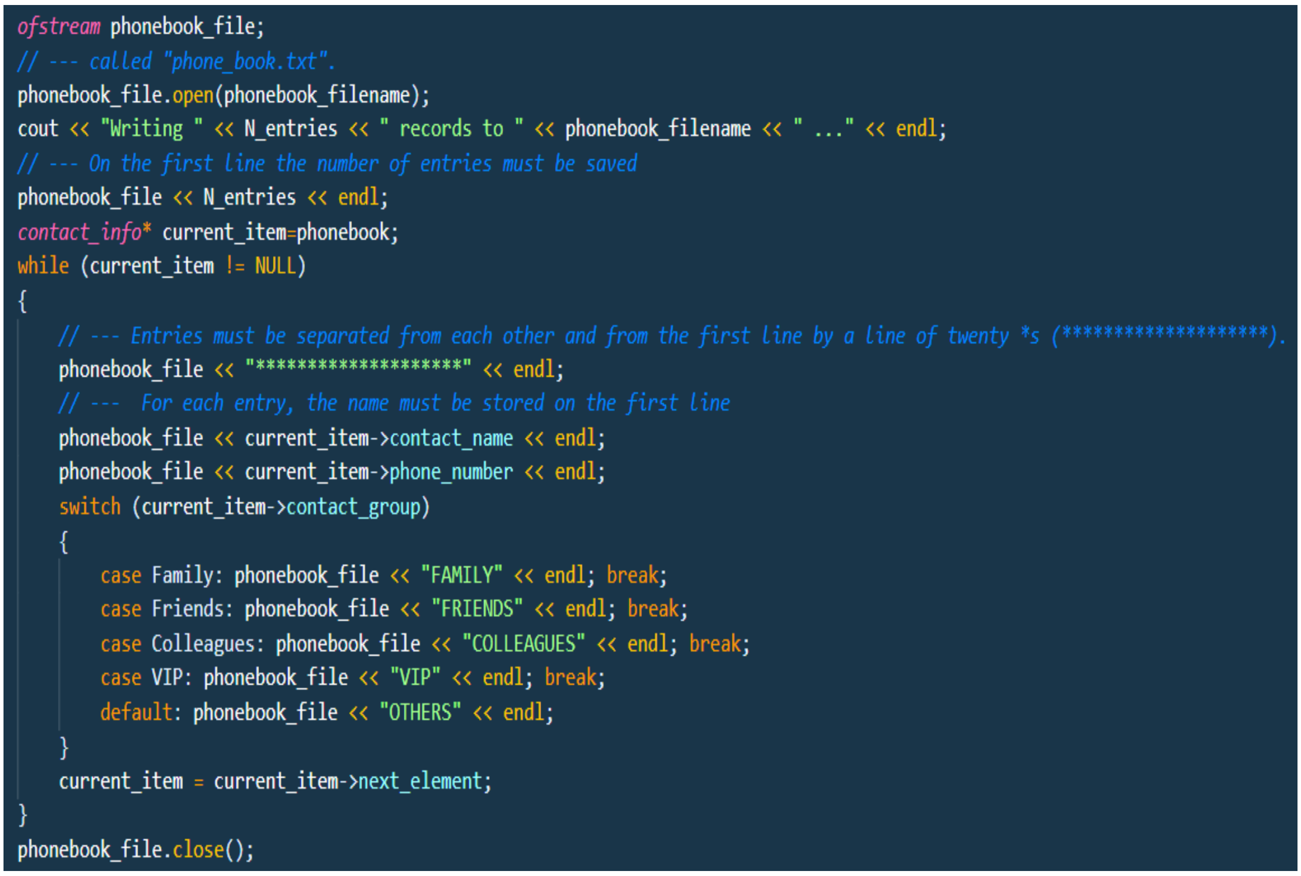
Save the whole phonebook in the file

This function of the program basically saves all the data of the contacts in file so that we can access later as user don’t want all the data to go away even after the compilation is done, so we made a .txt file and stores all the data in that only.



```
Please enter your choice: d
Writing 2 records to phone_book.txt ...
```

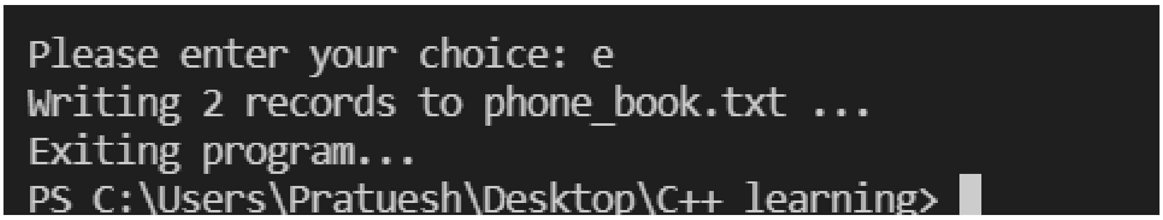
Code:-



```
ofstream phonebook_file;
// --- called "phone_book.txt".
phonebook_file.open(phonebook_filename);
cout << "Writing " << N_entries << " records to " << phonebook_filename << " ..." << endl;
// --- On the first line the number of entries must be saved
phonebook_file << N_entries << endl;
contact_info* current_item=phonebook;
while (current_item != NULL)
{
    // --- Entries must be separated from each other and from the first line by a line of twenty *s (********************).
    phonebook_file << "********************" << endl;
    // ---  For each entry, the name must be stored on the first line
    phonebook_file << current_item->contact_name << endl;
    phonebook_file << current_item->phone_number << endl;
    switch (current_item->contact_group)
    {
        case Family: phonebook_file << "FAMILY" << endl; break;
        case Friends: phonebook_file << "FRIENDS" << endl; break;
        case Colleagues: phonebook_file << "COLLEAGUES" << endl; break;
        case VIP: phonebook_file << "VIP" << endl; break;
        default: phonebook_file << "OTHERS" << endl;
    }
    current_item = current_item->next_element;
}
phonebook_file.close();
```

Exiting

This function of the program halts the execution of the code.



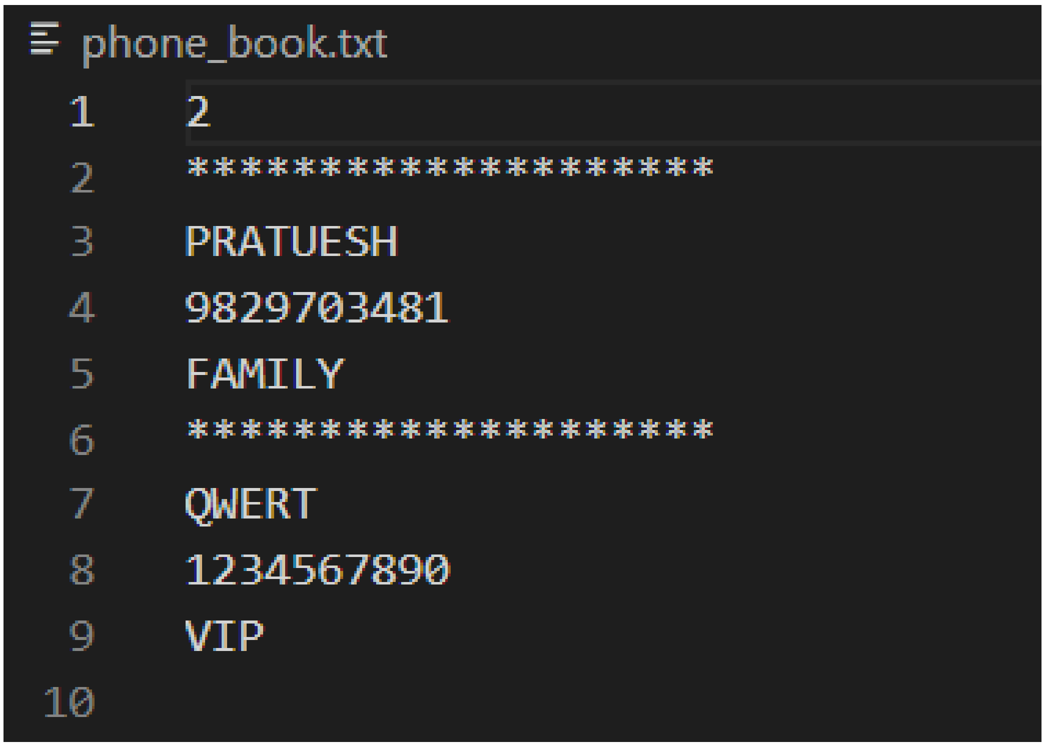
```
Please enter your choice: e
Writing 2 records to phone_book.txt ...
Exiting program...
PS C:\Users\Pratuesh\Desktop\C++ learning>
```

File:-

After you saves and exits from the program, all the data of the contacts are saved in the file. The file displays the number of entries after all the insertion and deletion. The file also displays the details of the contacts of the phonebook saved.

This saved file will be used to retrieve data to do the functions and will be updated according to the user.

Display of contacts in phonebook



phone_book.txt

```
2
********************
PRATUESH
9829703481
FAMILY
********************
QWERT
1234567890
VIP

```

Conclusion

Phone Book is a simple project help to us in that we get a simple solution to store our contacts. We can use it to replace our hard phonebook or even use it as an office wide phone directory. This will help user to easily search and manage contacts using this system. Phonebook is the one, which contain details of an individual along with their landline numbers. Apart from the telephone number of individuals, it also contains address and number of important relatives of individual. It not only contains local codes but also ISD codes.

For searching operation, users will able to get any particular record using their contact or phone number but the only condition is that, customers record must be available within the file system. If no such record will be available, proper error message will be displayed as per user input provided to the system.

The aim of this project was to build a Phonebook through which allowed to add, search, delete contacts of individual and access to database.

References

- https://www.tutorialandexample.com/advantage-disadvantage-linked-list/
- https://www.geeksforgeeks.org/data-structures/linked-list/#:~:text=A%20linked%20list%20is%20a,stored%20at%20contiguous%20memory%20locations.&text=In%20simple%20words%2C%20a%20linked,Singly%20Linked%20List
- https://www.javatpoint.com/singly-linked-list