



PHONE DIRECTORY

**Details of the Students:**

1. Amisha 9919103125
2. Rahul Malik 9919103126
3. Aviral Agarwal 9919103144

**Abstract:**

*Contacts Management* - We are going to implement the concept of hashing, hash table here by making a "Phone Directory" program that takes input which includes the user's name and their phone number and saves the data in the hash table. The hash table will use a hash function to compute an index into an array of buckets or slots, from which the correct value can be found.

Collision is the major problem in the hashing technique. In open addressing, all collisions are resolved in the prime area i.e., the area that contains all of the home addresses.

When a collision occurs, the prime area addresses are searches for an open or unoccupied element using linear probing.

Steps for inserting entities in a hash table:

1. If the location is empty, directly insert the entity.
2. If mapped location is occupied then keep probing until an empty slot is found. Once an empty slot is found, insert the entity.

We will insert the name of the person and the phone number related to that person. We create a hash table to store the data. We use insert function and, in that function, we will call the hash function. We declare the hash table also.

User can also create his own list of favourite contacts. The Phone Directory will help user to manage all the contacts and help them to retrieve contacts efficiently from the directory. User can easily insert, delete the contacts from directory.

A contact number retrieves contact’s details. A hash table is a perfect solution for this problem. As this allows you to get the response immediately. The user can also add most used contacts in favourites and can mark important contacts in the Directory.

Key Words: Hash Table, Arrays, Phone Directory, Contacts

**Tools and Techniques:**

* C++
* OOP
* File Handling
* Sorting Algorithms
* Hash Table
* GCC-32 bit

**Dataset Description:**

The Directory will contain all the social network contacts like various service helpline numbers, customer complaint numbers (by default). The dataset will also contain contact details of user’s social networks.

1. Create Record: It will take all the data inputs like the details of the contacts to be saved.
2. Display Record: It will display the details created.
3. Delete Record: It takes the key of the record to be deleted and then tries to search the key and if both matches, it deletes the record.
4. Search Record: It traverses the hash table, if record id matches with key, it displays the record details.
5. Add to favourites: It will allow to add the frequently connected contacts to new list.

**Design of the project:**

1. Log-in
2. Sign-up
3. Exit

3.Exit 2.Sign-up

Username

Password

Account created

1.Log-in

Username

Password

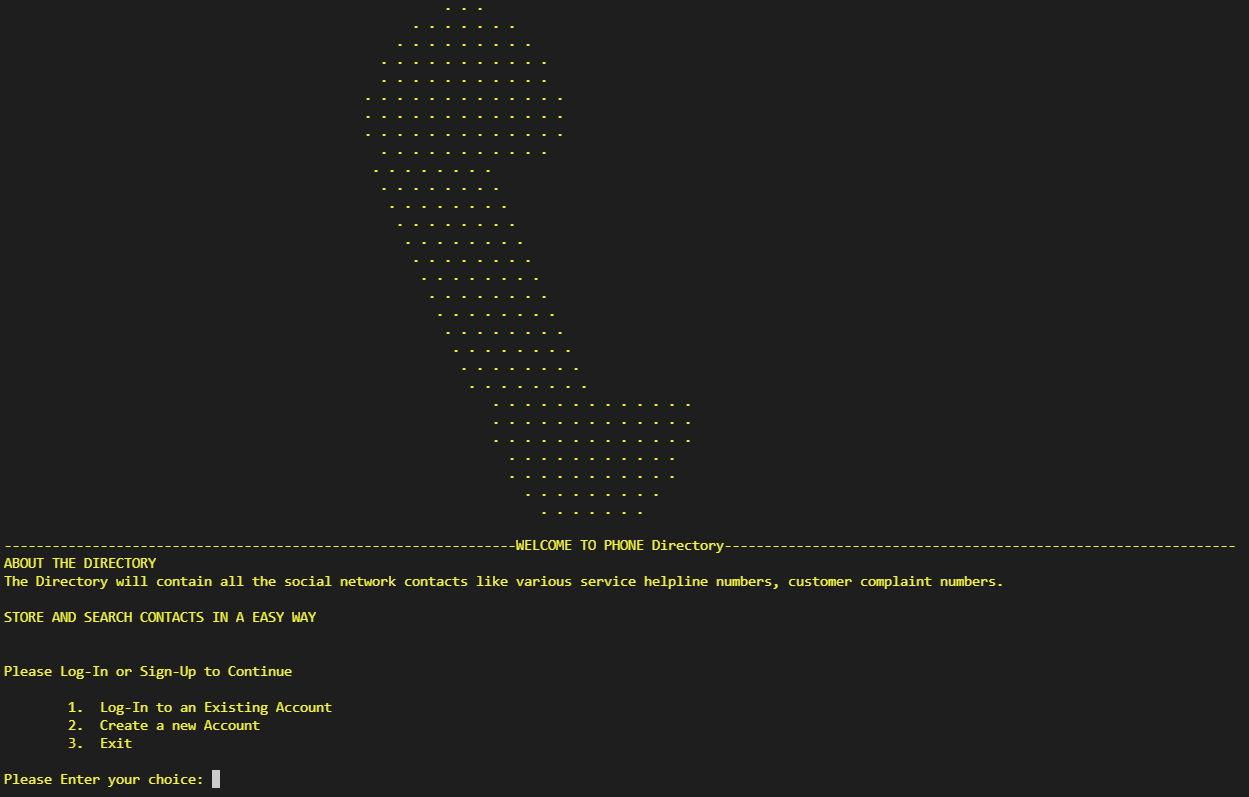
If User=Acc Member

If User = admin

1. Print all Contacts
2. Print all Accounts
3. Search Contacts
4. Enter New Contact
5. Delete Contacts
6. Modify Contacts
7. Log-out
8. Log-out & Exit
9. Print all Contacts
10. Search Contacts
11. Enter New Contact
12. Display Favorite Contacts
13. Log-out
14. Log-out & Exit

**Implementation details:**

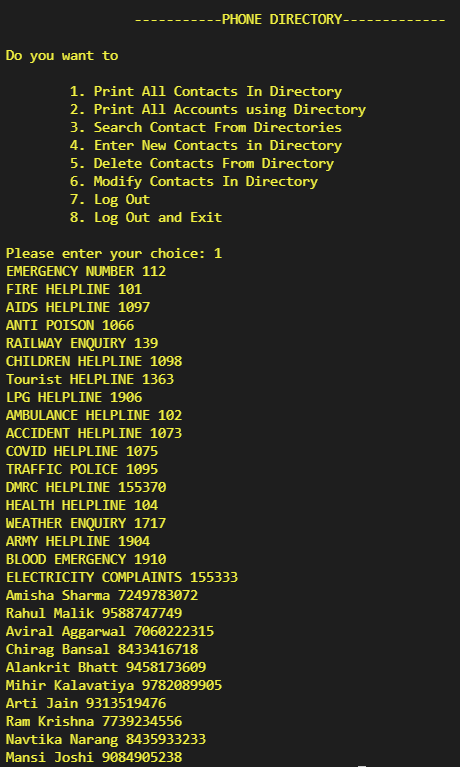
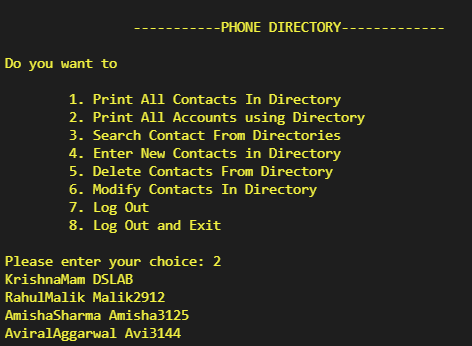
* Added pattern printing algorithms in program so that it visually looks good to user

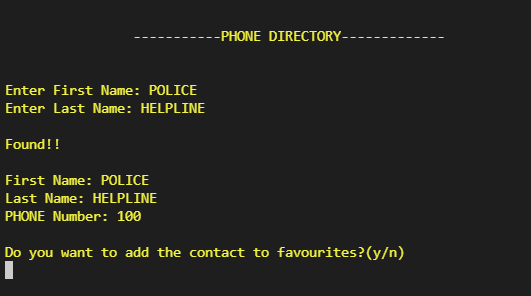


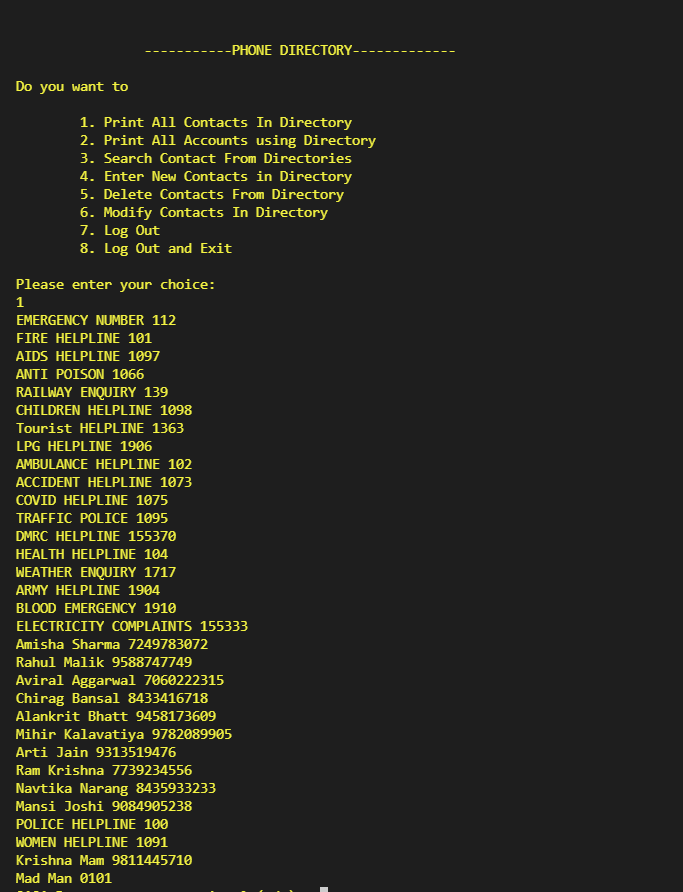
* Created a sign in and sign-up interface to keep the records of the accounts using the directory



* Added feature to search and display the phone number of any contact in the directory



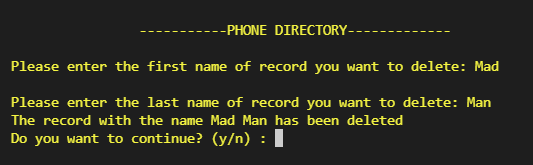


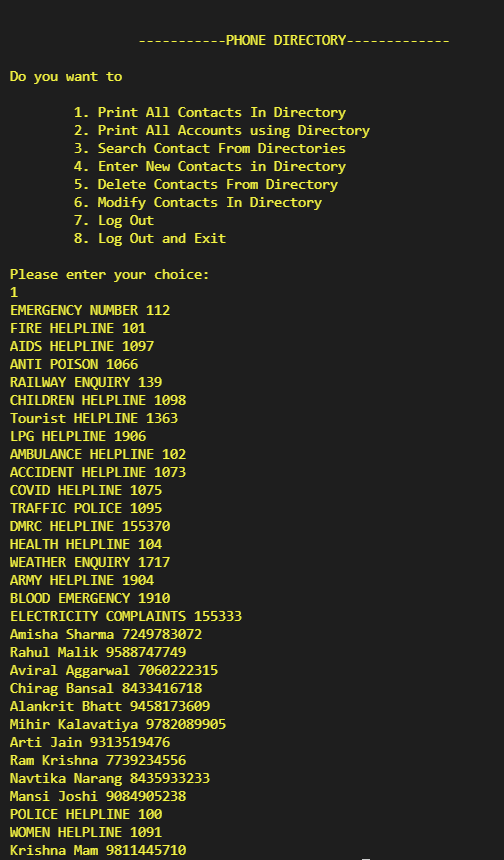


New Contact Mad Man added

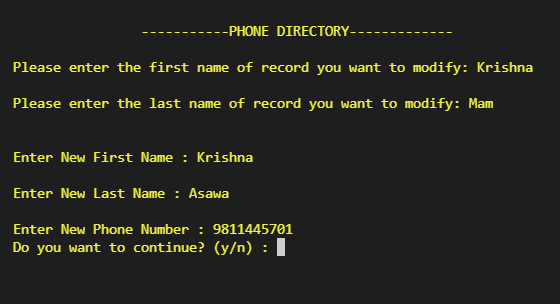
by some random user.

* So only admin has the authority to modify and delete a contact (for data security reasons).

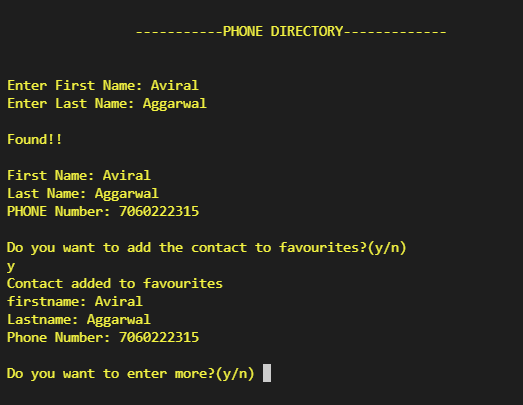


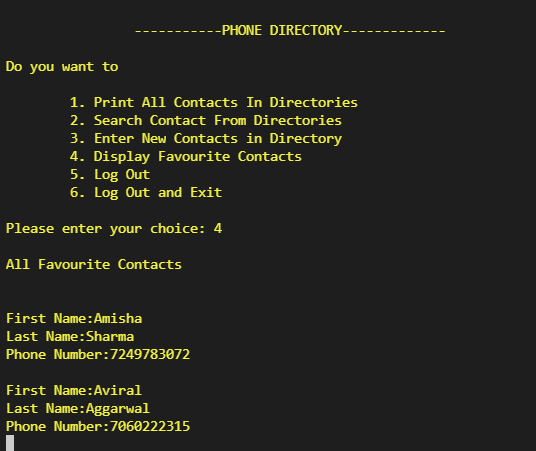


* To modify a contact in the directory using class function and OOPs.

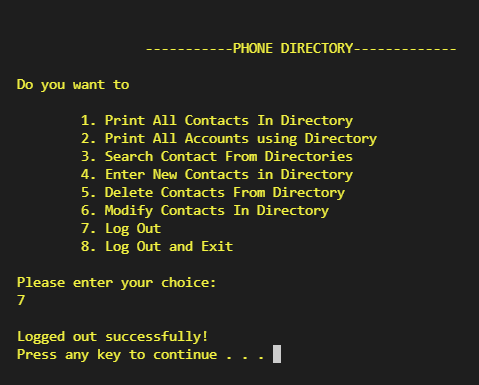


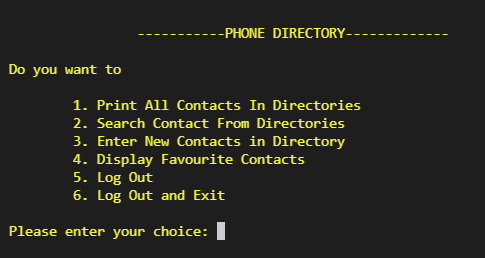
* Add to favorites: For adding contact to favorites list while adding new contact or displaying them, we have used Hashing method and stored them in a Hash Table. The hash function is used to determine the position in the hash table at which the value will be stored.





* Log-out





**References:**

* <https://www.geeksforgeeks.org/hashing-data-structure/>
* <https://www.geeksforgeeks.org/file-handling-c-classes/>
* <https://jogieglenmait.wordpress.com/2015/06/19/color-coding-for-c-console/>