Assignment No. 7

Q. Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client‘s telephone number.

```
#include <iostream>
#include <string>
using namespace std;

const int TABLE_SIZE = 10;

struct Client {
    string name;
    string phone;
    bool isOccupied;

    Client() {
        name = "";
        phone = "";
        isOccupied = false;
    }
};

class TelephoneBook {
    Client table[TABLE_SIZE];

    int hashFunction(string name) {
        int sum = 0;
        for (int i = 0; i < name.length(); i++)
            sum += name[i];
        return sum % TABLE_SIZE;
```

```
    }

    int searchIndex(string name) {
        int index = hashFunction(name);
        for (int i = 0; i < TABLE_SIZE; i++) {
            int probeIndex = (index + i) % TABLE_SIZE;
            if (table[probeIndex].isOccupied && table[probeIndex].name == name) {
                return probeIndex;
            }
        }
        return -1;
    }

public:
    void addClient(string name, string phone) {
        int index = hashFunction(name);
        for (int i = 0; i < TABLE_SIZE; i++) {
            int probeIndex = (index + i) % TABLE_SIZE;
            if (!table[probeIndex].isOccupied) {
                table[probeIndex].name = name;
                table[probeIndex].phone = phone;
                table[probeIndex].isOccupied = true;
                cout << "Added: " << name << " -> " << phone << endl;
                return;
            }
        }
        cout << "Telephone book is full. Cannot add " << name << "." << endl;
    }

    void getNumber(string name) {
        int idx = searchIndex(name);
        if (idx != -1)
```

```
            cout << name << "'s number is " << table[idx].phone << endl;
        else
            cout << name << " not found in telephone book." << endl;
    }

    void removeClient(string name) {
        int idx = searchIndex(name);
        if (idx != -1) {
            table[idx].isOccupied = false;
            table[idx].name = "";
            table[idx].phone = "";
            cout << "Removed: " << name << endl;
        } else {
            cout << name << " not found in telephone book." << endl;
        }
    }

    void displayAll() {
        cout << "\n--- Telephone Book Entries ---\n";
        for (int i = 0; i < TABLE_SIZE; i++) {
            if (table[i].isOccupied)
                cout << "[" << i << "] " << table[i].name << " -> " << table[i].phone << endl;
        }
    }
};

int main() {
    TelephoneBook book;

    book.addClient("Aarav", "9876543210");
    book.addClient("Ram", "9123456789");
    book.addClient("Rohan", "9988776655");
```

```
    book.addClient("Rupesh", "9001122334");
    book.addClient("Sahil", "9011223344");

    book.getNumber("Rohan");
    book.getNumber("Raj");

    book.displayAll();

    book.removeClient("Ram");
    book.displayAll();

    return 0;
}
```

Output :