Practical 6

Mayur Zope SE Comp A 75

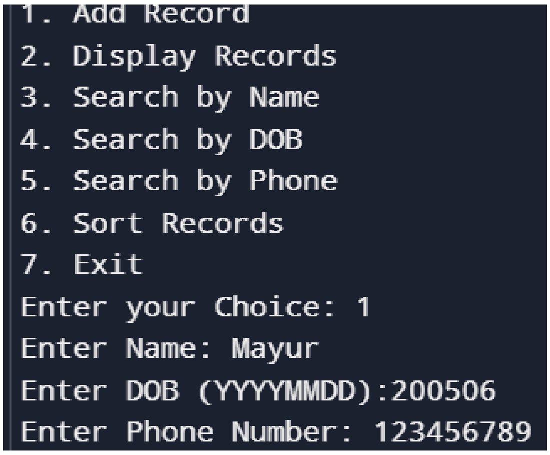
- **Write C++ program using STL for sorting and searching user defined records such as personal records (Name, DOB, Telephone number etc) using vector container.**

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>

using namespace std;

class PersonalRecord
{
        public:
                string name;
                string dob; // Date of Birth
                string phone;

                // Constructor
                PersonalRecord(string name, string dob, string phone)
                {
                        this->name = name;
                        this->dob = dob;
                        this->phone = phone;
                }
                // Display method
                void display() const
                {
                        cout<< "Name: "<<name<<", DOB: "<<dob<<", Phone: "<<phone<<endl;
                }
                // Comparison operator for sorting by name
        bool operator < (const PersonalRecord& other) const
        {
            return name < other.name;
        }
};

// Function to search for a record by name
PersonalRecord* searchByName (vector<PersonalRecord> &records, const string& name)
{
    for (auto& record : records)
    {
        if (record.name == name)
            return &record;
    }
    return nullptr;
}

// Function to search for a record by phone number
PersonalRecord* searchByPhone (vector<PersonalRecord> &records, const string& phone)
{
    for (auto& record : records)
    {
        if (record.phone == phone)
            return &record;
    }
    return nullptr;
}

// Function to search for a record by DOB
```

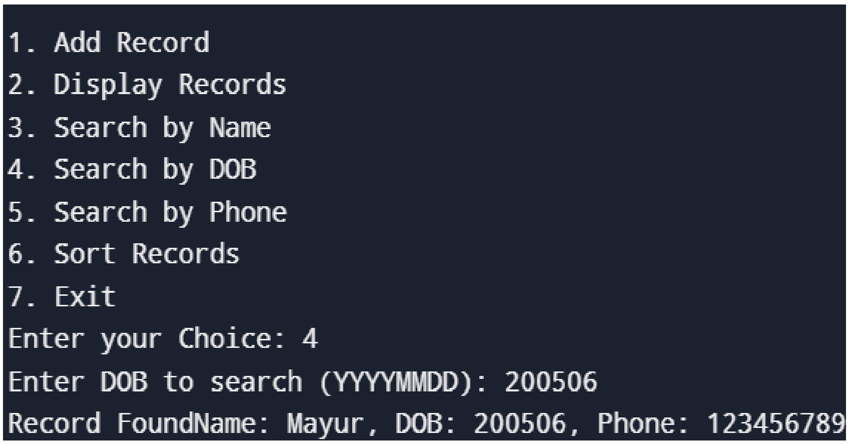
```
PersonalRecord* searchByDob (vector<PersonalRecord> &records, const string& dob)
{
    for (auto& record : records)
    {
        if (record.dob == dob)
            return &record;
    }
    return nullptr;
}

int main()
{
    vector <PersonalRecord> records;
    int choice;

    do {
        cout<<"\n1. Add Record\n2. Display Records\n3. Search by Name\n4. Search by DOB\n5. Search by Phone\n6. Sort Records\n7. Exit\n";
        cout<<"Enter your Choice: ";
        cin>>choice;
        cin.ignore();

        switch(choice)
        {
            case 1:
            {
                string name, dob, phone;
                cout<<"Enter Name: ";
                getline(cin, name);
                cout<<"Enter DOB (YYYYMMDD):";
                cin>>dob;
                cout<<"Enter Phone Number: ";
                cin>>phone;
                records.emplace_back(name, dob, phone);  // Add
new record
                break;
            }
            case 2:
            {
                cout<<"All Records:\n";
                for (const auto& record : records)
                {
                    record.display();  // Display each record
                }
                break;
            }
            case 3:
            {
                string name;
                cout<<"Enter name to  search: ";
                //cin.ignore();
                getline(cin, name);
                PersonalRecord* foundRecord =
searchByName(records, name);
                if (foundRecord)
                {
                    cout<<"Record Found";
                    foundRecord->display(); // Display the found
record
                }
```

```
                    else
                        cout<<"Record  not found.\n";
                    break;
                }
                case 4:
                {
                    string dob;
                    cout<<"Enter DOB to search (YYYYMMDD): ";
                    cin>>dob;
                    PersonalRecord* foundRecord = 
searchByDob(records, dob);
                    if (foundRecord)
                    {
                        cout<<"Record Found";
                        foundRecord->display(); // Display the found 
record
                    }
                    else
                        cout<<"Record  not found.\n";
                    break;


                }
                case 5:
                {
                    string phone;
                    cout<<"Enter phone number to  search: ";
                    cin>>phone;
                    PersonalRecord* foundRecord = 
searchByPhone(records, phone);
                    if (foundRecord)
                    {
```

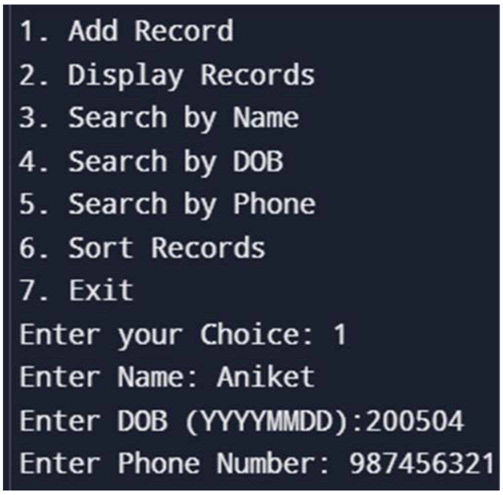
```
                    cout<<"Record Found";
                    foundRecord->display();  // Display the found
record
                }
                else
                    cout<<"Record  not found.\n";
                break;
            }
            case 6:
            {
                sort(records.begin(), records.end());  // Sort records
by name1
                cout<<"Records sorted by name.\n";
                break;
            }
            case 7:
            {
                cout<<"Exiting...\n";
                break;
            }
            default:
                cout<<"Invalid choice. Please try again.\n";
        }
    }
    while(choice != 7);

    return 0;
}
// OUTPUT
```



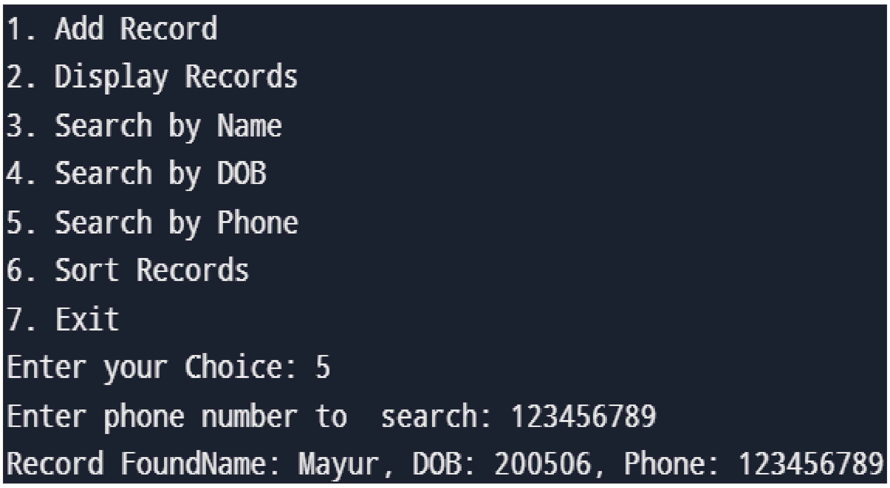
```
1. Add Record
2. Display Records
3. Search by Name
4. Search by DOB
5. Search by Phone
6. Sort Records
7. Exit
Enter your Choice: 1
Enter Name: Mayur
Enter DOB (YYYYMMDD):200506
Enter Phone Number: 123456789
```



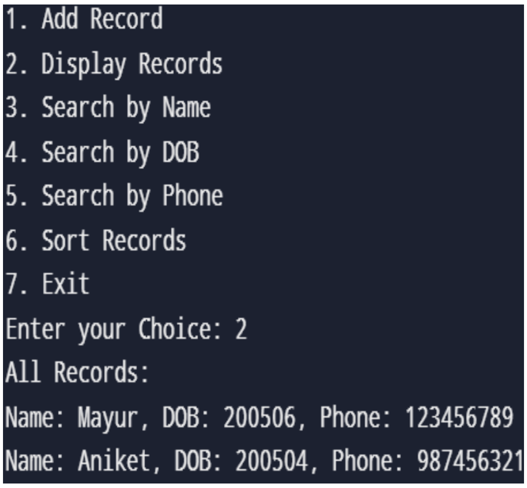
```
1. Add Record
2. Display Records
3. Search by Name
4. Search by DOB
5. Search by Phone
6. Sort Records
7. Exit
Enter your Choice: 4
Enter DOB to search (YYYYMMDD): 200506
Record FoundName: Mayur, DOB: 200506, Phone: 123456789
```



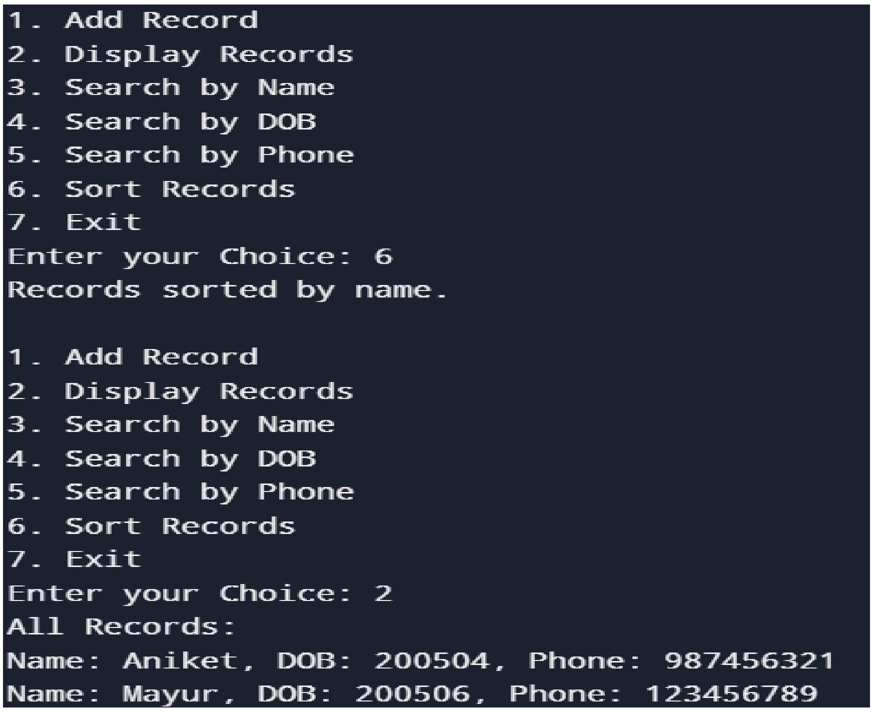
```
1. Add Record
2. Display Records
3. Search by Name
4. Search by DOB
5. Search by Phone
6. Sort Records
7. Exit
Enter your Choice: 1
Enter Name: Aniket
Enter DOB (YYYYMMDD):200504
Enter Phone Number: 987456321
```



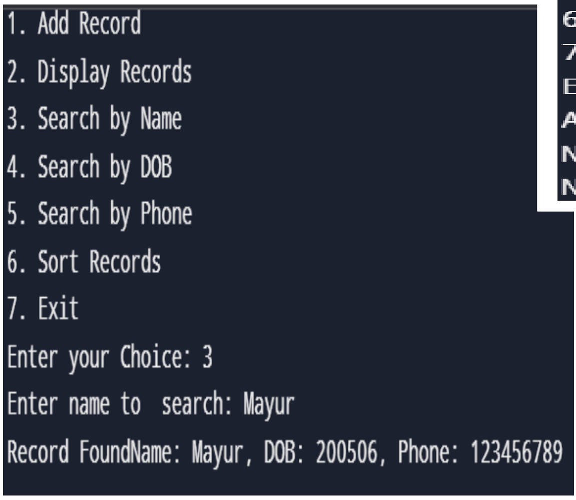
```
1. Add Record
2. Display Records
3. Search by Name
4. Search by DOB
5. Search by Phone
6. Sort Records
7. Exit
Enter your Choice: 5
Enter phone number to  search: 123456789
Record FoundName: Mayur, DOB: 200506, Phone: 123456789
```



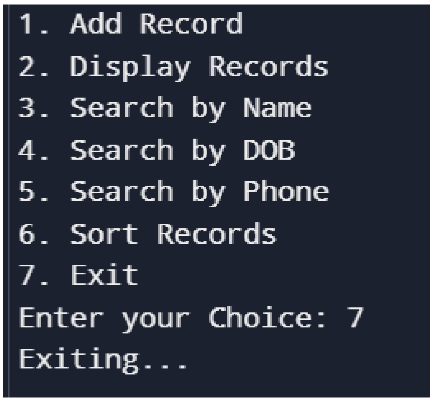
```
1. Add Record
2. Display Records
3. Search by Name
4. Search by DOB
5. Search by Phone
6. Sort Records
7. Exit
Enter your Choice: 2
All Records:
Name: Mayur, DOB: 200506, Phone: 123456789
Name: Aniket, DOB: 200504, Phone: 987456321
```



```
1. Add Record
2. Display Records
3. Search by Name
4. Search by DOB
5. Search by Phone
6. Sort Records
7. Exit
Enter your Choice: 6
Records sorted by name.

1. Add Record
2. Display Records
3. Search by Name
4. Search by DOB
5. Search by Phone
6. Sort Records
7. Exit
Enter your Choice: 2
All Records:
Name: Aniket, DOB: 200504, Phone: 987456321
Name: Mayur, DOB: 200506, Phone: 123456789
```



```
1. Add Record
2. Display Records
3. Search by Name
4. Search by DOB
5. Search by Phone
6. Sort Records
7. Exit
Enter your Choice: 3
Enter name to  search: Mayur
Record FoundName: Mayur, DOB: 200506, Phone: 123456789
```



```
1. Add Record
2. Display Records
3. Search by Name
4. Search by DOB
5. Search by Phone
6. Sort Records
7. Exit
Enter your Choice: 7
Exiting...
```