#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

struct PersonalRecord {

string name;

string dob; // Date of Birth in format YYYY-MM-DD

string phone;

// Overload < operator for sorting by name

bool operator<(const PersonalRecord &other) const {

return name < other.name;

}

};

void printRecords(const vector<PersonalRecord>& records) {

for (size\_t i = 0; i < records.size(); ++i) {

cout << "Name: " << records[i].name << ", DOB: " << records[i].dob << ", Phone: " << records[i].phone << endl;

}

}

bool searchRecord(const vector<PersonalRecord>& records, const string& name) {

for (size\_t i = 0; i < records.size(); ++i) {

if (records[i].name == name) {

cout << "Record found: " << "Name: " << records[i].name << ", DOB: " << records[i].dob << ", Phone: " << records[i].phone << endl;

return true;

}

}

cout << "Record not found." << endl;

return false;

}

int main() {

vector<PersonalRecord> records;

int n;

cout << "Enter the number of records: ";

cin >> n;

cin.ignore(); // Ignore newline left in the buffer

for (int i = 0; i < n; ++i) {

PersonalRecord record;

cout << "Enter details for record " << i + 1 << endl;

cout << "Name: ";

getline(cin, record.name);

cout << "DOB (YYYY-MM-DD): ";

getline(cin, record.dob);

cout << "Phone: ";

getline(cin, record.phone);

records.push\_back(record);

}

cout << "Original records:" << endl;

printRecords(records);

// Sort records by name

sort(records.begin(), records.end());

cout << "\nSorted records:" << endl;

printRecords(records);

// Search for a record by name

string nameToSearch;

cout << "\nEnter a name to search: ";

getline(cin, nameToSearch);

searchRecord(records, nameToSearch);

    return 0;

}