#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

using namespace std;

// Define a structure for personal records

struct PersonalRecord {

string name;

string dob; // Date of Birth

string telephone;

// Overload the < operator for sorting by name

bool operator<(const PersonalRecord& other) const {

return name < other.name;

}

};

// Function to display records

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

for (const auto& record : records) {

cout << "Name: " << record.name << ", DOB: " << record.dob << ", Telephone: " << record.telephone << endl;

}

}

// Function to find a record by name

void findRecord(const vector<PersonalRecord>& records, const string& name) {

auto it = find\_if(records.begin(), records.end(), [&name](const PersonalRecord& record) {

return record.name == name;

});

if (it != records.end()) {

cout << "Record found: Name: " << it->name << ", DOB: " << it->dob << ", Telephone: " << it->telephone << endl;

} else {

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

}

}

int main() {

vector<PersonalRecord> records;

// Adding sample records

records.push\_back({"Alice", "1990-01-01", "123-456-7890"});

records.push\_back({"Bob", "1985-05-23", "987-654-3210"});

records.push\_back({"Charlie", "1992-12-12", "555-555-5555"});

// Sorting records by name

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

// Display sorted records

cout << "Sorted Records:" << endl;

displayRecords(records);

// Find a record by name

string nameToFind;

cout << "Enter a name to find: ";

cin >> nameToFind;

findRecord(records, nameToFind);

return 0;

}