#include <iostream>

#include <vector>

#include <string>

#include <unordered\_map>

#include <unordered\_set>

using namespace std;

// Structure to hold course information

struct Course {

string name;

vector<string> availableTimeSlots;

vector<string> preferredRooms;

};

// Function to check if an assignment is valid

bool isValidAssignment(const string& timeSlot, const string& room,

const unordered\_map<string, pair<string, string>>& schedule) {

for (const auto& entry : schedule) {

if (entry.second.first == timeSlot && entry.second.second == room) {

return false; // Room already occupied at the time slot

}

}

return true;

}

// Function to schedule courses greedily

unordered\_map<string, pair<string, string>> scheduleCourses(vector<Course>& courses, vector<string>& timeSlots, vector<string>& rooms) {

unordered\_map<string, pair<string, string>> schedule;

for (const auto& course : courses) {

bool assigned = false;

for (const auto& timeSlot : course.availableTimeSlots) {

for (const auto& room : course.preferredRooms) {

if (isValidAssignment(timeSlot, room, schedule)) {

schedule[course.name] = {timeSlot, room};

assigned = true;

break;

}

}

if (assigned) break;

}

if (!assigned) {

cout << "Unable to schedule course: " << course.name << endl;

}

}

return schedule;

}

int main() {

// Input: Number of courses

int numCourses;

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

cin >> numCourses;

vector<Course> courses(numCourses);

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

cout << "Enter course name: ";

cin >> courses[i].name;

int numTimeSlots;

cout << "Enter the number of available time slots for " << courses[i].name << ": ";

cin >> numTimeSlots;

cout << "Enter available time slots: ";

for (int j = 0; j < numTimeSlots; ++j) {

string timeSlot;

cin >> timeSlot;

courses[i].availableTimeSlots.push\_back(timeSlot);

}

int numRooms;

cout << "Enter the number of preferred rooms for " << courses[i].name << ": ";

cin >> numRooms;

cout << "Enter preferred rooms: ";

for (int j = 0; j < numRooms; ++j) {

string room;

cin >> room;

courses[i].preferredRooms.push\_back(room);

}

}

// Input: Time slots and rooms

int numTimeSlots;

cout << "Enter the total number of time slots: ";

cin >> numTimeSlots;

vector<string> timeSlots(numTimeSlots);

cout << "Enter all time slots: ";

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

cin >> timeSlots[i];

}

int numRooms;

cout << "Enter the total number of rooms: ";

cin >> numRooms;

vector<string> rooms(numRooms);

cout << "Enter all rooms: ";

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

cin >> rooms[i];

}

// Schedule courses

auto schedule = scheduleCourses(courses, timeSlots, rooms);

// Output the schedule

cout << "Course Schedule:" << endl;

for (const auto& entry : schedule) {

cout << "Course: " << entry.first

<< ", Time Slot: " << entry.second.first

<< ", Room: " << entry.second.second << endl;

}

return 0;

}