#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

/\*

\* Example Graduate

\* Name: Jaired Jawed

\* Specialized Areas: Computer Science

\* Preferred Language: English

\* Preferred Language Fluency: .99

\*/

struct Graduate {

string name;

string specializedArea;

string preferredLanguage;

double preferredLanguageFluency; // Decimal from 0 to 1 that describes the Graduate's fluency

};

/\*

\* Example Country

\* Name: United Kingdom

\* Required Specialized Area: Computer Science

\* Language: English

\*/

struct Country {

string name;

string requiredSpecializedArea;

string language;

};

vector<Graduate> availableGraduates;

vector<Country> availableCountries;

/\* Function Headers \*/

void seedGraduates();

void seedCountries();

bool compareGraduateByLanguageFluency(Graduate, Graduate);

Country getSelectedCountry(string);

int main() {

seedGraduates();

seedCountries();

int i = 0;

cout << "Welcome to UCR's Doctors Without Borders Selection Program!" << endl;

for (; i < availableCountries.size(); i++) {

cout << i << ") " << availableCountries.at(i).name << endl;

}

string countryName;

cout << "Please enter a country to get the recommended graduates: ";

cin >> countryName;

// get the selected country by its name

Country selectedCountry = getSelectedCountry(countryName);

vector<Graduate> selectedGraduates;

for (i = 0; i < availableGraduates.size(); i++) {

Graduate currentGrad = availableGraduates.at(i);

// find all graduates that speak the selected country's language and required specialized area

if (selectedCountry.language == currentGrad.preferredLanguage && selectedCountry.requiredSpecializedArea == currentGrad.specializedArea) {

selectedGraduates.push\_back(currentGrad);

}

}

// Sorts the possible graduates for the selected countries by their fluency level

sort(selectedGraduates.begin(), selectedGraduates.end(), compareGraduateByLanguageFluency);

cout << "Here are the possible graduates for " << countryName << " by rank." << endl;

// List the selected graduates info

for (i = 0; i < selectedGraduates.size(); i++) {

Graduate currentGrad = selectedGraduates.at(i);

cout << "Name: " << currentGrad.name << endl;

cout << "Specialized Area: " << currentGrad.specializedArea << endl;

cout << "Preferred Language: " << currentGrad.preferredLanguage << endl;

cout << "Lanuage Fluency: " << currentGrad.preferredLanguageFluency << endl;

cout << endl;

}

cout << "End Program." << endl;

return 0;

}

/\* Push all graduates to vectors \*/

void seedGraduates() {

Graduate grad1;

grad1.name = "Alex";

grad1.preferredLanguage = "Tagalog";

grad1.specializedArea = "Heart Surgery";

grad1.preferredLanguageFluency = 0.71;

availableGraduates.push\_back(grad1);

Graduate grad2;

grad2.name = "Sandra";

grad2.preferredLanguage = "Tagalog";

grad2.specializedArea = "Heart Surgery";

grad2.preferredLanguageFluency = 0.98;

availableGraduates.push\_back(grad2);

Graduate grad3;

grad3.name = "David";

grad3.preferredLanguage = "English";

grad3.specializedArea = "Brain Surgery";

grad3.preferredLanguageFluency = 0.71;

availableGraduates.push\_back(grad3);

Graduate grad4;

grad4.name = "Vennessa";

grad4.preferredLanguage = "English";

grad4.specializedArea = "Brain Surgery";

grad4.preferredLanguageFluency = 0.91;

availableGraduates.push\_back(grad4);

Graduate grad5;

grad5.name = "Mark";

grad5.preferredLanguage = "German";

grad5.specializedArea = "Nursing";

grad5.preferredLanguageFluency = 0.64;

availableGraduates.push\_back(grad5);

Graduate grad6;

grad6.name = "Elizabeth";

grad6.preferredLanguage = "German";

grad6.specializedArea = "Nursing";

grad6.preferredLanguageFluency = 0.73;

availableGraduates.push\_back(grad6);

Graduate grad7;

grad7.name = "Leo";

grad7.preferredLanguage = "Spanish";

grad7.specializedArea = "Respitory Theripist";

grad7.preferredLanguageFluency = 0.56;

availableGraduates.push\_back(grad7);

Graduate grad8;

grad8.name = "Sarah";

grad8.preferredLanguage = "Spanish";

grad8.specializedArea = "Respitory Theripist";

grad8.preferredLanguageFluency = 0.43;

availableGraduates.push\_back(grad8);

Graduate grad9;

grad9.name = "Martin";

grad9.preferredLanguage = "Chinese";

grad9.specializedArea = "Neurology";

grad9.preferredLanguageFluency = 0.87;

availableGraduates.push\_back(grad9);

Graduate grad10;

grad10.name = "Susan";

grad10.preferredLanguage = "Chinese";

grad10.specializedArea = "Neurology";

grad10.preferredLanguageFluency = 0.96;

availableGraduates.push\_back(grad10);

}

/\* Push all countries to vectors \*/

void seedCountries() {

Country country1;

country1.name = "Phillipines";

country1.language = "Tagalog";

country1.requiredSpecializedArea = "Heart Surgery";

availableCountries.push\_back(country1);

Country country2;

country2.name = "Mexico";

country2.language = "Spanish";

country2.requiredSpecializedArea = "Respitory Theripist";

availableCountries.push\_back(country2);

Country country3;

country3.name = "Germany";

country3.language = "German";

country3.requiredSpecializedArea = "Nursing";

availableCountries.push\_back(country3);

Country country4;

country4.name = "USA";

country4.language = "English";

country4.requiredSpecializedArea = "Brain Surgery";

availableCountries.push\_back(country4);

Country country5;

country5.name = "China";

country5.language = "Chinese";

country5.requiredSpecializedArea = "Neurology";

availableCountries.push\_back(country5);

}

/\* Sorts Graduates by the greatest fluency level \*/

bool compareGraduateByLanguageFluency(const Graduate grad1, const Graduate grad2) {

return grad1.preferredLanguageFluency > grad2.preferredLanguageFluency;

}

/\* Find the Country by its name \*/

Country getSelectedCountry(string countryName) {

int i = 0;

for (; i < availableCountries.size(); i++) {

if (countryName == availableCountries.at(i).name) {

return availableCountries.at(i);

}

}

return availableCountries.at(0);

}