Team: Velvet Room

Members: Alberto Cervantes

The main purpose of this program is to allow people participating in the Doctors without Borders program to be systematically assigned to a country in need of their specific medical practices as well as based on their preferred language. This system of checking a user’s medical practice and a language they are fluent in allows for appropriate deployment to countries where people with the needed skills and necessary linguistic skills are needed. This program, while limited in the number of languages supported as well as number of supported medical practices, is able to assign participants to proper locations where their abilities can be used to their full potential.

When displayed the languages and medical specialties are ordered alphabetically through implementation of a selection sort function as well as a function which displays them numbered in their respective orders. The selection sort algorithm was selected as it is generally faster than a binary search and thus can be processed relatively quickly. Following this, the input validation comes in the form of checking that the user has not input invalid values into certain variables and allows them to input another value, these validations continue to loop until the user has input a valid option.

Major limitations of this program come in the form of how many languages are supported as well as the number of medical practices which are supported in the program. In addition to this, if a similar structure were kept for a larger scale version of this program, the program could likely slow down as sorting through and displaying each item, whether it be the languages supported or the medical practices supported, would take an even longer amount of time than this smaller-scale program can show. However, improvements to the limitations of the program are possible, and may be done through changing the selection sort to another method which can sort larger arrays much faster. In addition, an option to allow multiple users to use the program sequentially could be implemented with an option to exit the program with a simple yes or no input from the user after their results have been displayed. Possibly adding a data storage function to the program may be useful for maintaining records of what users have previously input into the program.

**Pseudo-code:**

// Pseudocode for Case 2: UCR Medical Center Volunteers

// Alberto Cervantes

// CIS 7 - Discrete Structures

// Dr. Nguyen

// https://www.doctorswithoutborders.org/what-we-do/countries#alphabetical-listing

List global variables / initialize some global variables for arrays [array sizes, languages, and specialties]

list possible participant specialties (minimum 5) in arrays.

[ Pediatrics, General Surgery, Neurology, Hematology, Orthopedic Surgery, Dietetics]

list languages of participants (minimum 7) as variables.

[ Hindi, English, Spanish, Arabic, Afrikaans, Italian, Portuguese, French]

sort functions for languages and specialties

display functions for sorted languages and specialties.

Begin main

variables for the user's name, specialty, and language.

Simple splash screen/ title.

Welcome user to Doctors without Borders program with short cout statement

ask user for information regarding name, language and specialty

// have their name recorded for later use

ask user for their name.

store name in variables.

// have ONE language the user is fluent in to minimize how much needs to be processed

ask user for a language they are fluent in (list preferred languages

[see list of languages listed in code above])

-store language in variable.

// have specialty of user to determine where they will go

ask user for medical specialty (list specialties)

-store specialty of user in variable.

// after checking for the language and specialty, the user will be given a list

// of possible deployment countries

case switch checking first for fluent language, followed by specialty.

// Will list all possible countries before switch statement of specialty for each language in comments.

Switch: (language check)

case 1:

Switch: (specialty check)

case switches for specialties (6)

show possible countries user can go to

case 2:

Switch: (specialty check)

case switches for specialties (6)

show possible countries user can go to

case 3:

Switch: (specialty check)

case switches for specialties (6)

show possible countries user can go to

case 4:

Switch: (specialty check)

case switches for specialties (6)

show possible countries user can go to

case 5:

Switch: (specialty check)

case switches for specialties (6)

show possible countries user can go to

case 6:

Switch: (specialty check)

case switches for specialties (6)

show possible countries user can go to

case 7:

Switch: (specialty check)

case switches for specialties (6)

show possible countries user can go to

case 8:

Switch: (specialty check)

case switches for specialties (6)

show possible countries user can go to

ending cout statement

End main

sort language function

sort specialty function

display functions for language and specialties respectively.

A close up of text on a white background

Description automatically generated