1. Team Name: Team Kimo
   Members: Christopher Kimoliatis

2. Project Information and details:
   a. This project solves the need of randomizing a selection of country for the doctors based on their choice of language and medical specializations. The program also shows the probability of the selection based on the selections of the doctors.
   b. The solution I am implementing in this project is to take inputs on their language preference and medical specialization from the doctors and having the program randomly select a country based on these inputs while calculating the probability of the selections.
   c. The objective of the program is to randomly select a country based on the choices made by the doctor or the user of the program. The user has two inputs, the language preference, and the medical specialization. These inputs are then used as variables in selecting a suitable country.
   d. Discrete structures are implemented in finding the probability based on the sample spaces in the program. The probability is then calculated and printed out in the code.
   e. This program has many limitations such as:
      i. This program was written in a way that is very static, where the user is only allowed to choose the languages and specializations provided. Although the Doctor's Without Borders program is a program where the countries are already set, the code itself is lacking in that it is not dynamic.
      ii. The randomization in the program may cause issues in a sense that if the randomizer randomly chooses the same countries, this will cause for an abundant number of doctors in one country whereas other countries will have less doctors.
   f. Recommendations in improving the limitations of this program:
      i. Instead of having the countries, languages, and medical specialties hard coded in a vector. It can be improved by creating classes that represent the countries, languages, and medical specialties. This can allow for a more dynamic program were adding and removing some countries or some languages is simpler.
      ii. Create a global counter that only allows a set number of doctors to go to a certain country. This creates a more evened out dispersion of doctors in the selected countries.

3. Pseudocode

```
Initialize chooseLanguage function header
Initialize chooseSpecialty funciton header

Function main
    Initialize 2 dimensional vector, countries holding the countries
    Initialize 1 dimensional vector, language holding all languages
    Call chooseLanguage function and initialize returned value
    Initialize 1 dimensional vector, specialties holding the specialties
    Call chooseSpecialty function and initialize returned value
    Output thanks
    Output the chosen language and chooseSpecialty
    Use srand(time(0)) to randomize the srand function to randomizes value
    Intialize n with a random integer with the range of countries
    Output chosen country based on randomized integer
    Initialize numberOfLanguage holding number of countries based on language
chosen
    Initialize numberOfSpecialties to hold the number of medical specialties

    Output the total possible outcomes
    Output the probability of matching countries
    Output the probability of matching specialties
    Output the total probability based on country, specialty, and language
choice
End main function

chooseLanguage function passing in language vector
    intialize languageChoice
    initialize numLanguage to number of elements in language vector
    Do
        Output choose language to user
        For i = 0 to number of langauges
            Output list of langauges
        Input languageChoice
        if languageChoice larger than numLanguage or smaller than 1
            Output try again
    Until languageChoice smaller than numLanguage or larger than 1
    return languageChoice
End chooseLanguage function

chooseSpecialty function passing in specialty vector
    Intialize specialization
    Do
        Output choose medical specialization to user
        For i=0 to number of medical specialties
            Output list of specialties
        Input specialization
        If specialization larger than 8 or larger than 1
            Output try again
    Until specialization smaller than numLanguage or larger than 1
    return specialization
End chooseSpecialty function
```