**Medical Diagnosis Expert System**

|  |  |
| --- | --- |
| *Shafayat Hossain*  *Electrical & Computer Engineering*  *ID: 2421387642*  *shafayat.hossain.242@northsouth.edu* | *Jannatun Noor Neel*  *Electrical & Computer Engineering*  *ID: 2423254642*  *jannatun.noor.242@northsouth.edu* |
| *Melissa Atiha*  *Electrical & Computer Engineering*  *ID: 2422813642*  *melissa.atiha.242@northsouth.edu* | *Saamin Yasar Hasan*  *Electrical & Computer Engineering*  *ID: 2422297642*  *saamin.hasan.242@northsouth.edu* |
|  |  |

**Introduction**

The goal of Medical Diagnosis Expert System is to help people recognize their symptoms and offer an initial diagnosis. This system is designed to identify the probability of two common diseases, Malaria and Dengue, by analyzing user answers to a predefined list of questions regarding symptoms.  
  
The system aims to increase knowledge about these diseases, helping early identification and direct users to seek medical advice. Its goal is to connect the difference between recognizing symptoms and receiving a diagnosis from an expert in a way that is easy for the user.

**Objective**

The primary objectives of this expert system are:

1. To provide a digital tool capable of evaluating symptoms of Malaria and Dengue.
2. To enable users to understand their symptoms better through prepared interactions.
3. To promote healthcare awareness by identifying overlapping symptoms that require further medical consultation.

**Code Analysis**

The program is coded in C, making use of its effectiveness for programming at a beginner level. A breakdown of the primary parts is provided below:

**Libraries used**

1. **<stdio.h>**: Handles input and output operations, such as printf() and scanf() for interacting with users
2. **<string.h>**: Used for string manipulation (not used extensively in this specific program).
3. **<ctype.h>**: Provides functions like tolower() to handle character input regardless of case.
4. **<math.h>**: Enables mathematical operations like abs(), which is used to evaluate the absolute difference between the scores of Malaria and Dengue.
5. **<unistd.h>**: Used for system-dependent operations like sleep() for a better user experience (e.g., timed messages).

**Key components of the code**

**Symptom Data**

Two arrays, dengue\_symptoms and malaria\_symptoms, store predefined patterns indicating whether each symptom is common for Dengue or Malaria.

A list of 14 symptom names (Symptoms[]) is used to ask questions interactively.

**Function**

* CalculateScore:

This function compares the user’s responses to the predefined patterns for both diseases.

For each symptom, if the user answered "yes" (y) and it matches the predefined symptom, the score for that disease is incremented.

* Diagnose:

This function interprets the scores:

1. If one score is significantly higher, the function indicates the user might have that condition.
2. If both scores are similar and moderately high, the program suggests overlapping symptoms and advises a medical checkup.
3. If both scores are low, the user is considered healthy.

**Error Handling and Continuous Workflow**

The program makes sure to only accept valid inputs (y or n). If it identifies an incorrect input, the program will end with an error message.  
  
Following the conclusion of a single diagnostic session, the user has the option to either restart the procedure or close the program.

**Advantages of the Program**

Interactive: Offers a user-friendly experience with explicit prompts and directions.  
  
Logical Analysis: Utilizes a robust decision-making algorithm for scoring calculations.

Iterative and adaptable: Permits users to carry out numerous sessions without needing to restart the software.

Educational: Improves user knowledge of Malaria and Dengue symptoms.

**Limitations**

User reliance: Accurate responses from the user are necessary for accurate results.  
  
Not medically validated: This tool is not a replacement for expert medical diagnosis.  
  
Restricted Focus: Only examines Malaria and Dengue, with other diseases presenting similar symptoms omitted from consideration.

**Conclusion**

This project showcases how C programming can be applied in practical healthcare situations. This program effectively demonstrates how technology can aid in initial health evaluation by combining user feedback, data analysis, and logical decision-making. Despite its restrictions, the tool functions as a learning assistance and establishes the basis for more advanced diagnostic systems.