

TRƯỜNG ĐẠI HỌC BẠC LIÊU
KHOA KỸ THUẬT & CÔNG NGHỆ



PHÁT TRIỂN ỨNG DỤNG TRÊN LINUX

TIÊU LUẬN
CHƯƠNG TRÌNH CHẨN ĐOÁN BỆNH

NHÓM: BÌNH MINH

GVHD: TS.NGÔ VIỆT THỊNH

THÀNH VIÊN

MSSV

Nguyễn Trường Khang **227480201058**

Ngô Hán Kiा **227480201060**

Võ Tấn Phát **227480201072**

Nguyễn Thanh Nguyên **227480201069**

Bạc Liêu, 12/2024

MỤC LỤC

I.Phần mở đầu	1
II. Phương pháp thực hiện	1
1. Đặt vấn đề (Mục tiêu, I/O).....	1
2. Thuật toán (Thuật toán và thư viện sử dụng)	3
2.1 Đọc triệu chứng đầu vào:	3
2.2 Đọc dữ liệu bệnh từ file:	3
2.3 So khớp triệu chứng:.....	3
2.4 Tính toán độ tin cậy:.....	3
2.5 Sắp xếp kết quả:.....	3
2.6 Hiển thị thông tin điều trị:	3
III. Kết quả	4
1.Code.....	4
2. Thuật toán Nhập Dữ Liệu và Lưu Trữ	19
3.giao diện	20
4.Github	23
5.google doc.....	23
IV. Kết luận	23

I.Phản mở đầu

Dự án "Chẩn đoán bệnh" được thực hiện trong khuôn khổ khóa học về hệ điều hành Linux nhằm phát triển một chương trình hỗ trợ người dùng trong việc chẩn đoán các bệnh thông qua các triệu chứng đã được nhập vào. Mục tiêu của dự án là xây dựng một hệ thống đơn giản nhưng hiệu quả, giúp người dùng có thể nhận diện các bệnh thông qua một số triệu chứng phổ biến mà họ gặp phải.

Chương trình hoạt động trên môi trường Linux và được phát triển bằng ngôn ngữ lập trình C. Nó sử dụng các kỹ thuật cơ bản trong hệ điều hành Linux như quản lý file, thao tác với dữ liệu từ file và xử lý chuỗi ký tự. Dữ liệu về bệnh và triệu chứng được lưu trữ trong các file văn bản, và chương trình sẽ đọc dữ liệu từ các file này để thực hiện việc chẩn đoán.

Chương trình có thể phân tích các triệu chứng người dùng nhập vào, so sánh với cơ sở dữ liệu bệnh trong hệ thống, sau đó đưa ra kết quả chẩn đoán về bệnh có thể gặp phải. Ngoài ra, chương trình còn cung cấp thông tin về thuốc trị bệnh và lời khuyên cho người dùng từ một file dữ liệu riêng biệt, giúp người dùng có cái nhìn tổng quan về cách chữa trị và chăm sóc sức khỏe. Chương trình có thể được mở rộng với nhiều loại bệnh khác nhau, và việc mở rộng cơ sở dữ liệu sẽ giúp nâng cao độ chính xác của chẩn đoán.

II. Phương pháp thực hiện

1. Đặt vấn đề (Mục tiêu, I/O)

Mục tiêu của dự án "Chẩn đoán bệnh" là xây dựng một chương trình hỗ trợ người dùng chẩn đoán các bệnh tiềm năng dựa trên các triệu chứng mà họ nhập vào. Hệ thống có nhiệm vụ so sánh các triệu chứng được cung cấp bởi người dùng với cơ sở dữ liệu bệnh đã được lưu trữ trước, sau đó đưa ra chẩn đoán kèm theo các thông tin về thuốc trị bệnh và lời khuyên (nếu có). Chương trình được phát triển trên nền tảng hệ điều hành Linux bằng ngôn ngữ lập trình C, sử dụng các kỹ thuật thao tác file để quản lý dữ liệu.

Dữ liệu đầu vào và đầu ra:

Đầu vào (Input):

Người dùng nhập vào danh sách các triệu chứng mà họ gặp phải thông qua giao diện dòng lệnh (CLI). Chương trình cũng sẽ đọc dữ liệu bệnh và triệu chứng từ file **Tenbenh.txt**, đồng thời lấy thông tin thuốc trị và lời khuyên từ file **Thuoctri.txt**.

Ví dụ:

```
===== MENU =====
1. Xem danh sach benh
2. Them ho so benh nhan
3. Chuan doan trieu chung cua ban
4. Xem danh sach ho so
5. Thoat
=====
Nhap lua chon: 3
Nhap so luong trieu chung cua ban (toi da 10): 4
Trieu chung 1: Chay nuoc mui
Trieu chung 2: Ho
Trieu chung 3: Dau hong
Trieu chung 4: Sot nhe
```

Đầu ra (Output):

Chương trình hiển thị danh sách các bệnh có thể xảy ra dựa trên các triệu chứng mà người dùng đã nhập. Các bệnh được xếp hạng theo mức độ tin cậy, đồng thời hiển thị thông tin về thuốc trị và lời khuyên (nếu có).

Ví dụ:

```
Nhap so luong trieu chung cua ban (toi da 10): 4
Trieu chung 1: Chay nuoc mui
Trieu chung 2: Ho
Trieu chung 3: Dau hong
Trieu chung 4: Sot nhe
Nhap ma ho so cua ban: 3842
=====
===== KET QUA CHAN DOAN =====

Benh : cam lanh
So trieu chung trung khop: 4
Do tin cay : 100.00%
Thuoc tri : paracetamol, vitamin c, thuoc khang histamin
Loi khuyen : nghi ngoi, uong nuoc am, giu am co the.

Benh : viem mui ho hap
So trieu chung trung khop: 3
Do tin cay : 75.00%
Thuoc tri : thuoc khang sinh, thuoc giam dau
Loi khuyen : nghi ngoi, giu am co hong, suc mieng voi nuoc muoi am.
Luu chan doan thanh cong!
```

2. Thuật toán (Thuật toán và thư viện sử dụng)

Chương trình sử dụng thuật toán so sánh chuỗi để thực hiện việc đối chiếu triệu chứng mà người dùng nhập vào với các triệu chứng có trong cơ sở dữ liệu. Quy trình thực hiện như sau:

2.1 Đọc triệu chứng đầu vào:

- Người dùng nhập danh sách các triệu chứng qua giao diện dòng lệnh.
- Chương trình chuẩn hóa dữ liệu đầu vào bằng cách chuyển về chữ thường và loại bỏ khoảng trắng thừa.

2.2 Đọc dữ liệu bệnh từ file:

File **Tenbenh.txt** được sử dụng để lưu trữ tên bệnh và danh sách triệu chứng tương ứng. Dữ liệu này được đọc và lưu vào một mảng cấu trúc Disease.

2.3 So khớp triệu chứng:

- Đối với từng bệnh trong cơ sở dữ liệu, chương trình so sánh từng triệu chứng do người dùng nhập vào với triệu chứng của bệnh đó.
- Đếm số lượng triệu chứng trùng khớp giữa đầu vào và cơ sở dữ liệu.

2.4 Tính toán độ tin cậy:

Mỗi triệu chứng trùng khớp sẽ đóng góp 25% vào độ tin cậy.

Công thức tính độ tin cậy: **confidence = match_count * 25.0f;**

- Nếu có hơn 4 triệu chứng trùng khớp, độ tin cậy được giới hạn ở mức tối đa 100%.

2.5 Sắp xếp kết quả:

- Các bệnh được sắp xếp theo thứ tự giảm dần của độ tin cậy, sử dụng hàm qsort.

2.6 Hiển thị thông tin điều trị:

Sau khi xác định các bệnh phù hợp, chương trình tra cứu file **Thuoctri.txt** để tìm thông tin về thuốc trị và lời khuyên.

Nếu có thông tin, chương trình hiển thị kèm theo kết quả chẩn đoán. Nếu không tìm thấy, thông báo rằng không có dữ liệu điều trị.

Thư viện sử dụng:

- **stdio.h**: Đọc và ghi dữ liệu, bao gồm việc in ra màn hình (printf), nhập từ người dùng (scanf, fgets), và thao tác file (fopen, fclose).
- **string.h**: Xử lý chuỗi, bao gồm các hàm như strcmp (so sánh chuỗi), strcpy (sao chép chuỗi), và strcspn (xóa ký tự newline).
- **stdlib.h**: Sử dụng cho các hàm quản lý bộ nhớ, chuyển đổi kiểu, và sắp xếp (qsort).
- **cctype.h**: Hỗ trợ chuẩn hóa ký tự, như tolower để chuyển đổi chữ hoa thành chữ thường.

Các bước xử lý chính:

1. **Chuẩn hóa dữ liệu**: Dùng tolower và loại bỏ khoảng trắng dư thừa để đồng nhất dữ liệu.
2. **Quản lý cấu trúc dữ liệu**: Lưu trữ dữ liệu bệnh và triệu chứng trong mảng cấu trúc Disease, thông tin điều trị trong Treatment.
3. **So sánh và lọc dữ liệu**: Sử dụng thuật toán so sánh chuỗi để đối chiếu triệu chứng người dùng với cơ sở dữ liệu.

Tóm tắt phương pháp:

Chương trình kết hợp các kỹ thuật cơ bản như xử lý chuỗi, thao tác file, và thuật toán sắp xếp để tạo ra một hệ thống chẩn đoán bệnh đơn giản nhưng hiệu quả. Mặc dù không áp dụng thuật toán trí tuệ nhân tạo, hệ thống có khả năng mở rộng dễ dàng bằng cách cập nhật thêm cơ sở dữ liệu bệnh và thông tin điều trị. Phương pháp tiếp cận này phù hợp với mục tiêu phát triển trên môi trường Linux và sử dụng các công cụ cơ bản của ngôn ngữ C.

III. Kết quả

1. Code

```
#include <stdio.h>  
#include <string.h>
```

```

#include <stdlib.h>
#include <ctype.h>
#include <time.h>
#include "PatientRecord.h"

#define MAX_SYMPTOMS 10
#define MAX_DISEASES 100
#define MAX_PATIENTS 100
#define MAX_LINE_LENGTH 200
#define MAX_PATIENTS 100

struct PatientRecord patients[MAX_PATIENTS]; // Mảng chung cho bốn nhân
int patient_count = 0;

struct Treatment {
    char disease_name[50];
    char medicines[200];
    char advice[200];
};

// Cấu trúc bệnh
struct Disease {
    char name[50];
    char symptoms[MAX_SYMPTOMS][50];
    int symptom_count;
};

// Cấu trúc kết quả chẩn đoán
struct DiagnosisResult {

```

```

char disease_name[50];
int match_count;
float confidence;
};

// Hàm chuẩn hóa chuỗi
void normalize(char *str) {
    for (int i = 0; str[i]; i++) {
        str[i] = tolower(str[i]);
    }
}

void trim(char *str) {
    int i = 0, j = strlen(str) - 1;
    while (isspace(str[i])) i++;
    while (isspace(str[j])) j--;
    for (int k = 0; k <= j - i; k++) {
        str[k] = str[i + k];
    }
    str[j - i + 1] = '\0';
}

void normalize_and_trim(char *str) {
    normalize(str);
    trim(str);
}

int load_treatments_from_file(const char *filename, struct Treatment treatments[]) {
    FILE *file = fopen(filename, "r");

```

```

if (!file) {
    perror("Loi mo file");
    return 0;
}

int treatment_count = 0;
char line[MAX_LINE_LENGTH];
struct Treatment current_treatment = { "", "", "" };

while (fgets(line, sizeof(line), file)) {
    if (strlen(line) <= 1) continue;

    line[strcspn(line, "\n")] = '\0';
    normalize_and_trim(line);

    if (strncmp(line, "ten benh:", 9) == 0) {
        if (strlen(current_treatment.disease_name) > 0) {
            treatments[treatment_count++] = current_treatment;
            current_treatment = (struct Treatment) { "", "", "" };
        }
        strcpy(current_treatment.disease_name, line + 10);
    } else if (strncmp(line, "thuoc tri:", 10) == 0) {
        strcpy(current_treatment.medicines, line + 11);
    } else if (strncmp(line, "loi khuyen:", 11) == 0) {
        strcpy(current_treatment.advice, line + 12);
    }
}

```

```

if (strlen(current_treatment.disease_name) > 0) {
    treatments[treatment_count++] = current_treatment;
}

fclose(file);
return treatment_count;
}

// Hàm đ?c danh sách b?nh t? file
int load_diseases_from_file(const char *filename, struct Disease diseases[]) {
    FILE *file = fopen(filename, "r");
    if (!file) {
        perror("Loi mo file");
        return 0;
    }

    int disease_count = 0;
    char line[MAX_LINE_LENGTH];
    struct Disease current_disease = { "", {}, 0 };

    while (fgets(line, sizeof(line), file)) {
        if (strlen(line) <= 1) continue;

        line[strcspn(line, "\n")] = '\0';
        normalize_and_trim(line);

        if (strncmp(line, "ten benh:", 9) == 0) {
            if (current_disease.symptom_count > 0) {

```

```

        diseases[disease_count++] = current_disease;
        current_disease = (struct Disease) { "", {}, 0 };

    }

    strcpy(current_disease.name, line + 10);
    normalize_and_trim(current_disease.name);

} else if (strncmp(line, "trieu chung:", 12) == 0) {
    continue;

} else if (isdigit(line[0])) {

    char *symptom = strchr(line, '.');
    if (symptom != NULL) {

        symptom++;
        normalize_and_trim(symptom);
        strcpy(current_disease.symptoms[current_disease.symptom_count++], symptom);

    }
}

if (current_disease.symptom_count > 0) {

    diseases[disease_count++] = current_disease;
}

fclose(file);

return disease_count;
}

void find_treatment(const char *disease_name, struct Treatment treatments[], int treatment_count) {

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

if (strcmp(disease_name, treatments[i].disease_name) == 0) {

printf("%-25s: %s\n", "Thuoc tri", treatments[i].medicines);
}
}
}

```

```

        printf("%-25s: %s\n", "Loi khuyen", treatments[i].advice);

    return;
}

}

printf("%-25s: Khong co thong tin\n", "Thuoc tri");
printf("%-25s: Khong co thong tin\n", "Loi khuyen");

}

// Hàm t?o m? ng?u nhi?n 4 ký t? s?
void generate_random_id(char *id) {
    srand(time(NULL));
    sprintf(id, "%04d", rand() % 10000);
}

// Hàm luu h? so b?nh nh?n vào file
void save_patient_record() {
    if (patient_count >= MAX_PATIENTS) {
        printf("Danh sach ho so da day. Khong the them moi!\n");
        return;
    }

    struct PatientRecord record;

    // Nh?p thông tin h? so
    printf("Nh?p ten benh nhan: ");
    fgets(record.name, sizeof(record.name), stdin);
    record.name[strcspn(record.name, "\n")] = '\0';
}

```

```

// Nh?p gi?i tinh
printf("Nh?p gioi tinh (Nam/Nu): ");
fgets(record.gender, sizeof(record.gender), stdin);
record.gender[strcspn(record.gender, "\n")] = '\0'; // X? lý newline c?n l?i

printf("Nh?p nam sinh: ");
scanf("%d", &record.birth_year);
getchar();

record.age = 2024 - record.birth_year;

printf("Nh?p noi o: ");
fgets(record.address, sizeof(record.address), stdin);
record.address[strcspn(record.address, "\n")] = '\0';

// T?o m?u ng?u nhi?n
generate_random_id(record.id);

// Luu v?o m?ng
patients[patient_count++] = record;

printf("Luu ho so thanh cong! Ma ho so cua ban la: %s\n", record.id);
}

void view_patient_records() {
    if (patient_count == 0) {
        printf("Khong co ho so nao!\n");
}

```

```

        return;
    }

printf("\n===== DANH SACH HO SO =====\n");
for (int i = 0; i < patient_count; i++) {
    printf("Ma ho so: %s\n", patients[i].id);
    printf("Ten: %s\n", patients[i].name);
    printf("Gioi tinh: %s\n", patients[i].gender);
    printf("Nam sinh: %d\n", patients[i].birth_year);
    printf("Tuoi: %d\n", patients[i].age);
    printf("Noi o: %s\n", patients[i].address);
    printf("Chuan doan: %s\n", strlen(patients[i].diagnosis) > 0 ? patients[i].diagnosis : "Chua
co du lieu");
    printf("=====\\n");
}
}

// Hàm lưu k?t qu? ch?n đoán vào h? so
void save_diagnosis_to_record(const char *patient_id, const char *diagnosis) {
    for (int i = 0; i < patient_count; i++) {
        if (strcmp(patient_id, patients[i].id) == 0) {
            strcpy(patients[i].diagnosis, diagnosis);
            printf("Luu chan doan thanh cong!\\n");
            return;
        }
    }
    printf("Khong tim thay ho so voi ma: %s\\n", patient_id);
}

```

```
}
```

```
// Hàm hi?n th? menu
```

```
void display_menu() {
```

```
    printf("\n===== MENU =====\n");
```

```
    printf("1. Xem danh sach benh\n");
```

```
    printf("2. Them ho so benh nhan\n");
```

```
    printf("3. Chuan doan trieu chung cua ban\n");
```

```
    printf("4. Xem danh sach ho so\n");
```

```
    printf("5. Thoat\n");
```

```
    printf("=====\\n");
```

```
}
```

```
// Hàm ch?n doán b?nh (dã c?p nh?t k?t n?i luu h? so)
```

```
void diagnose(char input_symptoms[MAX_SYMPTOMS][50], int num_input_symptoms, struct Disease diseases[], int num_diseases, struct Treatment treatments[], int treatment_count) {
```

```
    char patient_id[5];
```

```
    printf("Nh?p ma ho so cua ban: ");
```

```
    fgets(patient_id, sizeof(patient_id), stdin);
```

```
    patient_id[strcspn(patient_id, "\\n")] = '\\0';
```

```
// Ki?m tra mÃ h? so
```

```
int found = 0;
```

```
for (int i = 0; i < patient_count; i++) {
```

```
    if (strcmp(patient_id, patients[i].id) == 0) {
```

```
        found = 1;
```

```
        break;
```

```

        }

    }

if (!found) {
    printf("Khong tim thay ma ho so: %s\n", patient_id);
    return;
}

struct DiagnosisResult results[MAX_DISEASES];
int result_count = 0;

// Ch?n doán b?nh
for (int i = 0; i < num_diseases; i++) {
    int match_count = 0;

    for (int j = 0; j < num_input_symptoms; j++) {
        normalize_and_trim(input_symptoms[j]);

        for (int k = 0; k < diseases[i].symptom_count; k++) {
            normalize_and_trim(diseases[i].symptoms[k]);

            if (strcmp(input_symptoms[j], diseases[i].symptoms[k]) == 0) {
                match_count++;
                break;
            }
        }
    }
}

```

```

if (match_count > 2) {
    float confidence = match_count * 25.0f;
    if (confidence > 100.0f) confidence = 100.0f;

    strcpy(results[result_count].disease_name, diseases[i].name);
    results[result_count].match_count = match_count;
    results[result_count].confidence = confidence;
    result_count++;
}

if (result_count > 0) {
    printf("\n===== KET QUA CHAN DOAN =====\n");
    char diagnosis[500] = "";
    for (int i = 0; i < result_count; i++) {
        printf("\n%-25s: %s\n", "Benh", results[i].disease_name);
        printf("%-25s: %d\n", "So trieu chung trung khop", results[i].match_count);
        printf("%-25s: %.2f%\n", "Do tin cay", results[i].confidence);

        find_treatment(results[i].disease_name, treatments, treatment_count);
        strcat(diagnosis, results[i].disease_name);
        strcat(diagnosis, "; ");
    }

    save_diagnosis_to_record(patient_id, diagnosis);
} else {
    printf("Khong tim thay benh phu hop.\n");
}

```

```
}
```

```
// Hàm chính
```

```
int main() {
```

```
    // Khai báo danh sách bệnh và danh sách thuốc
```

```
    struct Disease diseases[MAX_DISEASES];
```

```
    struct Treatment treatments[MAX_DISEASES];
```

```
    // Tạo dữ liệu từ file
```

```
    int num_diseases = load_diseases_from_file("Tenbenh.txt", diseases);
```

```
    int num_treatments = load_treatments_from_file("Thuoctri.txt", treatments);
```

```
    // Kiểm tra dữ liệu
```

```
    if (num_diseases == 0 || num_treatments == 0) {
```

```
        printf("Không có dữ liệu bệnh hoặc thuốc trong file.\n");
```

```
        return 1;
```

```
}
```

```
int choice; // Lấy chọn menu
```

```
do {
```

```
    // Hiển thị menu
```

```
    display_menu();
```

```
    // Nhập lựa chọn
```

```
    printf("Nhập lựa chọn: ");
```

```
    scanf("%d", &choice);
```

```

getchar(); // Xóa ký tự newline còn lại trong bộ nhớ

// Xử lý từng lựa chọn
switch (choice) {
    case 1: {
        // Xem danh sách bệnh
        printf("\nDanh sách các bệnh phổ biến:\n");
        for (int i = 0; i < num_diseases; i++) {
            printf("%-5d %-50s\n", i + 1, diseases[i].name);
        }
        break;
    }
    case 2: {
        // Thêm hồ sơ bệnh nhân
        save_patient_record();
        break;
    }
    case 3: {
        // Chẩn đoán triệu chứng
        char input_symptoms[MAX_SYMPTOMS][50];
        int num_input_symptoms;

        // Nhập số lượng triệu chứng của bạn (tối đa %d):
        printf("Nhập số lượng triệu chứng của bạn (tối đa %d): ", MAX_SYMPTOMS);
        scanf("%d", &num_input_symptoms);
        getchar();

        // Nhập tên triệu chứng

```

```

        for (int i = 0; i < num_input_symptoms; i++) {
            printf("Trieu chung %d: ", i + 1);
            fgets(input_symptoms[i], sizeof(input_symptoms[i]), stdin);
            input_symptoms[i][strcspn(input_symptoms[i], "\n")] = '\0'; // Loại bỏ newline
        }

        // Gọi hàm chẩn đoán
        diagnose(input_symptoms, num_input_symptoms, diseases, num_diseases, treatments,
        num_treatments);

        break;
    }

    case 4: {
        // Xem danh sách hồ sơ
        view_patient_records();
        break;
    }

    case 5: {
        // Thoát chương trình
        printf("Thoát chương trình...\n");
        break;
    }

    default: {
        // Xử lý lỗi nếu không hợp lệ
        printf("Lựa chọn không hợp lệ. Vui lòng thử lại!\n");
    }
}

} while (choice != 5); // Thoát khi người dùng chọn 5

return 0;

```

}

2. Thuật toán Nhập Dữ Liệu và Lưu Trữ

Dự án cũng bao gồm một chức năng cho phép người dùng nhập các triệu chứng của bệnh nhân và lưu trữ chúng vào mảng để tiến hành chẩn đoán:

```
#include <stdio.h>
```

```
void inputSymptoms(Diagnosis *disease) {  
    printf("Nhập triệu chứng: ");  
    fgets(disease->symptom, sizeof(disease->symptom), stdin);  
    printf("Nhập chuẩn đoán: ");  
    fgets(disease->diagnosis, sizeof(disease->diagnosis), stdin);  
}  
  
int main() {  
    Diagnosis symptoms[10];  
    int num = 0;  
  
    printf("Nhập số lượng bệnh nhân: ");  
    scanf("%d", &num);  
    getchar(); // Clear buffer  
  
    for (int i = 0; i < num; i++) {  
        inputSymptoms(&symptoms[i]);  
    }  
  
    diagnose(symptoms, num);  
  
    return 0;  
}
```

3.giao diện

```
===== MENU =====
1. Xem danh sach benh
2. Them ho so benh nhan
3. Chuan doan trieu chung cua ban
4. Xem danh sach ho so
5. Thoat
=====
Nhap lua chon: 1

Danh sach cac benh pho bien:
1   cam lanh
2   cam sot
3   viem phe quan
4   dau da day
5   tieu duong
6   viem hong
7   dau nho mat
8   suy gan
9   suy than
10  viem phoi
11  cao huyet ap
12  thap khop
13  quai bi
14  thieu mau
15  suy tim
16  tieu chay
17  dau rang
18  suy giap
19  cang thang
20  viem bang quang
21  viem ruot thua
22  viem amidan
23  viem gan b
24  hen suyen
25  lao phoi
26  viem xoang
27  viem khop dang thap
28  dau lung
29  benh gout
30  dau tai
31  loet da day
32  phong
33  mat ngu
34  dau co vai gay
35  dau bung kinh
36  rung toc
37  so gan
38  lao ruot
39  nhiem vi rut rota
40  benh sot ret
```

```
41 mat viem giac mac
42 ho hen
43 viem ket mac
44 viem duong ho hap
45 dau nguc trai
46 viem da dau
47 thieu mau do loet da day
48 viem tuyen nuoc bot
49 benh dau vai tot cap da lat
50 sot cap do gan day
51 viem nao
52 ung thu gan
53 ung thu phoi
54 ung thu da day
55 viem dai trang
56 tieu buon
57 suy nhuoc co the
58 benh liem phoi
59 viem tuy
60 nhiem hiv
61 quai do (rubella)
62 thuy dau
63 u xo tu cung
64 mat ngu
65 benh phong
66 viem loet mieng
67 suy giap tuyen giap
68 viem bao quy dau
69 lao phoi keo dai
70 benh vong dai trang
71 viem mui ho hap
72 tieu duong thai ky
73 bien chung tieu duong
74 hen phe quan
75 voi hoa dot song
76 benh vien nau
77 viem ruot
78 viem da co
79 viem tay chan miu
80 viem gan a
81 suy vanh hoa
82 di ung
83 doi loat huyet
84 viem dai trang
85 binh co cung
86 viem khop gai phau
87 bieu hien ung thu
88 viem phoi nguyen hinh
89 benh tim mach
90 cao huyet ap
91 loet da gai phau
```

```

90  cao huyet ap
91  loet da gai phau
92  nham vi rut ruot
93  ung thu tien xuong
94  gan hieu trung
95  viem giap
96  lao am
97  doi loat
98  benh lau
99  loet da song
100  viem da

===== MENU =====
1. Xem danh sach benh
2. Them ho so benh nhan
3. Chuan doan trieu chung cua ban
4. Xem danh sach ho so
5. Thoat
=====
Nhap lua chon: 2
Nhap ten benh nhan: Nguyen Van A
Nhap gioi tinh (Nam/Nu): Nam
Nhap nam sinh: 1998
Nhap noi o: Bac Lieu
Luu ho so thanh cong! Ma ho so cua ban la: 0565

===== MENU =====
1. Xem danh sach benh
2. Them ho so benh nhan
3. Chuan doan trieu chung cua ban
4. Xem danh sach ho so
5. Thoat
=====
Nhap lua chon: 3
Nhap so luong trieu chung cua ban (toi da 10): 4
Trieu chung 1: Chay nuoc mui
Trieu chung 2: Ho
Trieu chung 3: Dau hong
Trieu chung 4: Sot nhe
Nhap ma ho so cua ban: 0565

===== KET QUA CHAN DOAN =====
Benh : cam lanh
So trieu chung trung khop: 4
Do tin cay : 100.00%
Thuoc tri : paracetamol, vitamin c, thuoc khang histamin
Loi khuyen : nghỉ ngơi, uống nước ấm, giữ ấm cơ thể.

Benh : viem mui ho hap
So trieu chung trung khop: 3

```

```

90  cao huyet ap
91  loet da gai phau
92  nham vi rut ruot
93  ung thu tien xuong
94  gan hieu trung
95  viem giap
96  lao am
97  doi loat
98  benh lau
99  loet da song
100  viem da

===== MENU =====
1. Xem danh sach benh
2. Them ho so benh nhan
3. Chuan doan trieu chung cua ban
4. Xem danh sach ho so
5. Thoat
=====
Nhap lua chon: 2
Nhap ten benh nhan: Nguyen Van A
Nhap gioi tinh (Nam/Nu): Nam
Nhap nam sinh: 1998
Nhap noi o: Bac Lieu
Luu ho so thanh cong! Ma ho so cua ban la: 0565

===== MENU =====
1. Xem danh sach benh
2. Them ho so benh nhan
3. Chuan doan trieu chung cua ban
4. Xem danh sach ho so
5. Thoat
=====
Nhap lua chon: 3
Nhap so luong trieu chung cua ban (toi da 10): 4
Trieu chung 1: Chay nuoc mui
Trieu chung 2: Ho
Trieu chung 3: Dau hong
Trieu chung 4: Sot nhe
Nhap ma ho so cua ban: 0565

===== KET QUA CHAN DOAN =====
Benh : cam lanh
So trieu chung trung khop: 4
Do tin cay : 100.00%
Thuoc tri : paracetamol, vitamin c, thuoc khang histamin
Loi khuyen : nghỉ ngơi, uống nước ấm, giữ ấm cơ thể.

Benh : viem mui ho hap
So trieu chung trung khop: 3

```

4.Github

Đây là đường dẫn vào github: [Bấm vào đây](#)

5.google doc

IV. Kết luận

Dự án "Chuẩn Đoán Bệnh" đã chứng minh ứng dụng của các thuật toán cơ bản trong hệ thống chẩn đoán bệnh sử dụng các triệu chứng y tế phổ biến. Bằng cách triển khai thu thập và xử lý dữ liệu thông qua các thuật toán điều kiện đơn giản, hệ thống có thể xác định các bệnh có thể có từ danh sách triệu chứng. Dự án không chỉ cung cấp

các khái niệm về xử lý dữ liệu mà còn giúp làm rõ cách các thuật toán có thể được sử dụng trong các ứng dụng thực tế.

Về hiệu suất, hệ thống đã có thể xử lý nhiều đầu vào và cung cấp chẩn đoán phù hợp dựa trên thông tin người dùng nhập vào. Tuy nhiên, phương pháp này còn đơn giản và các mô hình học máy phức tạp hơn có thể cải thiện độ chính xác và tính linh hoạt của hệ thống chẩn đoán. Dự án này là nền tảng giúp hiểu cách công nghệ phần mềm có thể hỗ trợ trong ngành y tế qua các hệ thống phân tích và xử lý thông tin y tế.

Các câu hỏi để hỏi AI:

Báo cáo dự án (~5 trang) - Tiếng Anh & Tiếng Việt như sau: Phần giới thiệu (viết phần giới thiệu dài nửa trang cho một dự án "Chẩn đoán bệnh" trong khóa học về hệ điều hành Linux)

Methodology (sử dụng song ngữ tiếng Việt tiếng Anh)

Problem statement (Objective, I/O-1 page)

Algorithms (AI & Libraries-1 page)

Làm một chương trình C về chẩn đoán bệnh với đầu vào là file Tenbenh.txt trong file này chứa các loại bệnh và triệu chứng. yêu cầu chương trình chạy ra dự đoán 4 triệu chứng để cho kết quả bệnh.