#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Define the structure for Patient

typedef struct Patient {

char name[50];

int age;

char disease[50];

char admission\_date[20];

} Patient;

// Queue node structure

typedef struct Node {

Patient patient;

struct Node\* next;

} Node;

Node\* front = NULL;

Node\* rear = NULL;

// Function to enqueue a patient

void enqueue(Patient patient) {

Node\* new\_node = (Node\*)malloc(sizeof(Node));

new\_node->patient = patient;

new\_node->next = NULL;

if (rear == NULL) {

front = rear = new\_node;

} else {

rear->next = new\_node;

rear = new\_node;

}

printf("Patient added to the queue.\n");

}

// Function to dequeue a patient

void dequeue() {

if (front == NULL) {

printf("No patients in the queue.\n");

return;

}

Node\* temp = front;

printf("Serving Patient:\nName: %s\nAge: %d\nDisease: %s\nAdmission Date: %s\n",

temp->patient.name, temp->patient.age, temp->patient.disease, temp->patient.admission\_date);

front = front->next;

if (front == NULL) {

rear = NULL;

}

free(temp);

}

// Function to display all patients in the queue

void displayQueue() {

if (front == NULL) {

printf("No patients in the queue.\n");

return;

}

Node\* temp = front;

printf("Patients in the queue:\n");

while (temp != NULL) {

printf("Name: %s, Age: %d, Disease: %s, Admission Date: %s\n",

temp->patient.name, temp->patient.age, temp->patient.disease, temp->patient.admission\_date);

temp = temp->next;

}

}

// Save queue to file

void saveToFile() {

FILE\* file = fopen("patients\_data.txt", "w");

if (file == NULL) {

printf("Error saving to file.\n");

return;

}

Node\* temp = front;

while (temp != NULL) {

fprintf(file, "%s %d %s %s\n", temp->patient.name, temp->patient.age, temp->patient.disease, temp->patient.admission\_date);

temp = temp->next;

}

fclose(file);

printf("Data saved to file.\n");

}

// Load queue from file

void loadFromFile() {

FILE\* file = fopen("patients\_data.txt", "r");

if (file == NULL) {

printf("No previous data found.\n");

return;

}

Patient patient;

while (fscanf(file, "%s %d %s %s", patient.name, &patient.age, patient.disease, patient.admission\_date) != EOF) {

enqueue(patient);

}

fclose(file);

printf("Data loaded from file.\n");

}

int main() {

loadFromFile(); // Load data from file if available

int choice;

do {

printf("\nHospital Management System\n");

printf("1. Add Patient\n");

printf("2. Serve Patient\n");

printf("3. Display Patients\n");

printf("4. Save Data\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1: {

Patient patient;

printf("Enter name: ");

scanf("%s", patient.name);

printf("Enter age: ");

scanf("%d", &patient.age);

printf("Enter disease: ");

scanf("%s", patient.disease);

printf("Enter admission date: ");

scanf("%s", patient.admission\_date);

enqueue(patient);

break;

}

case 2:

dequeue();

break;

case 3:

displayQueue();

break;

case 4:

saveToFile();

break;

case 5:

saveToFile(); // Save data before exiting

printf("Exiting...\n");

break;

default:

printf("Invalid choice. Try again.\n");

}

} while (choice != 5);

return 0;

}