#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct rendez\_vous {

char jour[50];

int heure;

};

struct patient {

char prenom[50];

char nom[50];

char date\_naissance[15];

struct rendez\_vous rdv;

struct patient \*suivant;

};

struct médecin {

char nom[50];

char adresse[50];

char spécialité[50];

char ville[50];

int disponibilité[7][10];

struct patient \*file\_rendez\_vous;

struct médecin \*suivant;

};

struct médecin\* ajouter\_médecin(struct médecin \*tête) {

struct médecin \*nouveau = (struct médecin \*)malloc(sizeof(struct médecin));

if (nouveau == NULL) {

printf("Échec d'allocation mémoire\n");

return tête;

}

getchar();

printf("Donner le nom du médecin :\n");

fgets(nouveau->nom, sizeof(nouveau->nom), stdin);

nouveau->nom[strcspn(nouveau->nom, "\n")] = '\0';

printf("Donner l'adresse du médecin :\n");

fgets(nouveau->adresse, sizeof(nouveau->adresse), stdin);

nouveau->adresse[strcspn(nouveau->adresse, "\n")] = '\0';

printf("Donner la spécialité du médecin :\n");

fgets(nouveau->spécialité, sizeof(nouveau->spécialité), stdin);

nouveau->spécialité[strcspn(nouveau->spécialité, "\n")] = '\0';

printf("Donner la ville du médecin :\n");

fgets(nouveau->ville, sizeof(nouveau->ville), stdin);

nouveau->ville[strcspn(nouveau->ville, "\n")] = '\0';

printf("Saisir les disponibilités du médecin %s (0 = non disponible, 1 = disponible) :\n", nouveau->nom);

char jours[7][10] = {"Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche"};

for (int j = 0; j < 7; j++) {

printf("\n%s :\n", jours[j]);

for (int k = 0; k < 10; k++) {

printf(" Heure %d:00 (0=indisponible, 1=disponible) : ", 8 + k);

scanf("%d", &nouveau->disponibilité[j][k]);

while (nouveau->disponibilité[j][k] != 0 && nouveau->disponibilité[j][k] != 1) {

printf("Saisie invalide. Entrez 0 ou 1 pour l'heure %d:00 : ", 8 + k);

scanf("%d", &nouveau->disponibilité[j][k]);

}

}

}

nouveau->suivant = tête;

return nouveau;

}

void afficher\_disponibilités(struct médecin \*méd) {

const char \*jours[7] = {"Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche"};

printf("\nDisponibilités du médecin %s :\n", méd->nom);

for (int j = 0; j < 7; j++) {

printf("%s : ", jours[j]);

for (int k = 0; k < 10; k++) {

printf("%d:00 - %s ", 8 + k, méd->disponibilité[j][k] == 1 ? "Disponible" : "Indisponible");

}

printf("\n");

}

}

void afficherRendezVous(struct médecin \*méd) {

printf("\nRendez-vous pour le médecin %s :\n", méd->nom);

struct patient \*current\_patient = méd->file\_rendez\_vous;

if (current\_patient == NULL) {

printf("Aucun rendez-vous enregistré.\n");

return;

}

while (current\_patient != NULL) {

printf("Patient : %s %s, Date de naissance : %s\n", current\_patient->prenom, current\_patient->nom, current\_patient->date\_naissance);

printf("Jour du rendez-vous : %s, Heure : %d:00\n", current\_patient->rdv.jour, current\_patient->rdv.heure);

current\_patient = current\_patient->suivant;

}

}

void ajouterPatient(struct médecin \*med, struct patient \*nouveau\_patient) {

struct patient \*temp = med->file\_rendez\_vous;

if (temp == NULL) {

med->file\_rendez\_vous = nouveau\_patient;

} else {

while (temp->suivant != NULL) {

temp = temp->suivant;

}

temp->suivant = nouveau\_patient;

}

nouveau\_patient->suivant = NULL;

}

void prendreRendezVous(struct médecin \*tête) {

struct patient \*nouveau\_patient = malloc(sizeof(struct patient));

if (!nouveau\_patient) {

printf("Échec de l'allocation mémoire\n");

return;

}

printf("Entrez votre prénom : ");

getchar();

fgets(nouveau\_patient->prenom, sizeof(nouveau\_patient->prenom), stdin);

nouveau\_patient->prenom[strcspn(nouveau\_patient->prenom, "\n")] = '\0';

printf("Entrez votre nom : ");

fgets(nouveau\_patient->nom, sizeof(nouveau\_patient->nom), stdin);

nouveau\_patient->nom[strcspn(nouveau\_patient->nom, "\n")] = '\0';

printf("Entrez votre date de naissance (jj/mm/aaaa) : ");

fgets(nouveau\_patient->date\_naissance, sizeof(nouveau\_patient->date\_naissance), stdin);

nouveau\_patient->date\_naissance[strcspn(nouveau\_patient->date\_naissance, "\n")] = '\0';

char nom\_med[50];

printf("Entrez le nom du médecin : ");

fgets(nom\_med, sizeof(nom\_med), stdin);

nom\_med[strcspn(nom\_med, "\n")] = '\0';

struct médecin \*courant = tête;

while (courant != NULL) {

if (strcmp(courant->nom, nom\_med) == 0) {

printf("Disponibilités pour le médecin %s :\n", courant->nom);

afficher\_disponibilités(courant);

printf("Choisissez un jour : ");

fgets(nouveau\_patient->rdv.jour, sizeof(nouveau\_patient->rdv.jour), stdin);

nouveau\_patient->rdv.jour[strcspn(nouveau\_patient->rdv.jour, "\n")] = '\0';

printf("Choisissez une heure (8-18) : ");

scanf("%d", &nouveau\_patient->rdv.heure);

int jour\_index = -1;

const char \*jours[7] = {"Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche"};

for (int j = 0; j < 7; j++) {

if (strcmp(jours[j], nouveau\_patient->rdv.jour) == 0) {

jour\_index = j;

break;

}

}

if (jour\_index != -1 && courant->disponibilité[jour\_index][nouveau\_patient->rdv.heure - 8] == 1) {

courant->disponibilité[jour\_index][nouveau\_patient->rdv.heure - 8] = 0;

ajouterPatient(courant, nouveau\_patient);

printf("Rendez-vous confirmé avec le médecin %s à %d:00 le %s\n", courant->nom, nouveau\_patient->rdv.heure, nouveau\_patient->rdv.jour);

} else {

printf("Créneau indisponible, veuillez réessayer.\n");

free(nouveau\_patient);

}

return;

}

courant = courant->suivant;

}

printf("Médecin non trouvé.\n");

free(nouveau\_patient);

}

void sauvegarder\_médecins(struct médecin \*tête) {

FILE \*fichier = fopen("medecins.txt", "w");

if (fichier == NULL) {

printf("Erreur d'ouverture du fichier de sauvegarde.\n");

return;

}

struct médecin \*actuel = tête;

while (actuel != NULL) {

fprintf(fichier, "%s\n", actuel->nom);

fprintf(fichier, "%s\n", actuel->adresse);

fprintf(fichier, "%s\n", actuel->spécialité);

fprintf(fichier, "%s\n", actuel->ville);

for (int i = 0; i < 7; i++) {

for (int j = 0; j < 10; j++) {

fprintf(fichier, "%d ", actuel->disponibilité[i][j]);

}

fprintf(fichier, "\n");

}

struct patient \*p = actuel->file\_rendez\_vous;

while (p != NULL) {

fprintf(fichier, "%s %s %s %s %d\n", p->prenom, p->nom, p->date\_naissance, p->rdv.jour, p->rdv.heure);

p = p->suivant;

}

fprintf(fichier, "\n");

actuel = actuel->suivant;

}

fclose(fichier);

printf("Données sauvegardées avec succès.\n");

}

int main() {

struct médecin \*tête = NULL;

int choix;

do {

printf("\nMenu :\n");

printf("1. Ajouter un médecin\n");

printf("2. Prendre un rendez-vous\n");

printf("3. Afficher les rendez-vous d'un médecin\n");

printf("4. Sauvegarder les données\n");

printf("5. Quitter\n");

printf("Entrez votre choix : ");

scanf("%d", &choix);

switch (choix) {

case 1:

tête = ajouter\_médecin(tête);

break;

case 2:

prendreRendezVous(tête);

break;

case 3:

afficherRendezVous(tête);

break;

case 4:

sauvegarder\_médecins(tête);

break;

case 5:

printf("Au revoir !\n");

break;

default:

printf("Choix invalide, réessayez.\n");

}

} while (choix != 5);

return 0;

}