/\*

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_AD 20

#define MAX\_KAYIT 100

typedef struct Hasta {

int hastaNo;

char ad[MAX\_AD];

int yas;

char cinsiyet;

int mahkum;

float engellilik;

int kanamali;

char kayitSaat[6];

char muayeneSaat[6];

int muayeneSure;

int oncelik;

struct Hasta\* next;

} Hasta;

Hasta\* ilkHasta = NULL;

Hasta\* sonHasta = NULL;

Hasta\* kayitOlanlar = NULL;

int saatleriTopla(char\* saat1, char\* saat2) {

int saat1\_saat = atoi(strtok(saat1, ":"));

int saat1\_dakika = atoi(strtok(NULL, ":"));

int saat2\_saat = atoi(strtok(saat2, ":"));

int saat2\_dakika = atoi(strtok(NULL, ":"));

int toplam\_saat = saat1\_saat + saat2\_saat;

int toplam\_dakika = saat1\_dakika + saat2\_dakika;

if (toplam\_dakika >= 60) {

toplam\_saat += 1;

toplam\_dakika -= 60;

}

return (toplam\_saat \* 100) + toplam\_dakika;

}

void SiraliEkle(Hasta\* hasta) {

if (ilkHasta == NULL) {

ilkHasta = sonHasta = hasta;

return;

}

Hasta\* iter = ilkHasta;

Hasta\* prev = NULL;

while (iter != NULL && hasta->oncelik <= iter->oncelik) {

prev = iter;

iter = iter->next;

}

if (iter == ilkHasta) {

hasta->next = ilkHasta;

ilkHasta = hasta;

} else if (iter == NULL) {

prev->next = hasta;

hasta->next = NULL;

sonHasta = hasta;

} else {

prev->next = hasta;

hasta->next = iter;

}

}

void HastalariYukle() {

FILE\* dosya = fopen("hasta.txt", "r");

if (dosya == NULL) {

printf("Dosya açılamadı.\n");

return;

}

char satir[100];

while (fgets(satir, sizeof(satir), dosya) != NULL) {

Hasta\* hasta = (Hasta\*)malloc(sizeof(Hasta));

hasta->hastaNo = -1; // Hasta numarası atanacak

sscanf(satir, "%s %d %c %d %f %d %s %d", hasta->ad, &hasta->yas, &hasta->cinsiyet,

&hasta->mahkum, &hasta->engellilik, &hasta->kanamali, hasta->kayitSaat,

&hasta->muayeneSure);

hasta->oncelik = OncelikPuaniHesapla(hasta);

strcpy(hasta->muayeneSaat, hasta->kayitSaat);

hasta->next = NULL;

SiraliEkle(hasta);

}

fclose(dosya);

}

void HastaNoEkle() {

Hasta\* iter = ilkHasta;

int hastaNo = 1;

while (iter != NULL) {

iter->hastaNo = hastaNo;

hastaNo++;

iter = iter->next;

}

}

void Yazdir() {

Hasta\* iter = ilkHasta;

while (iter != NULL) {

printf("Hasta No: %d\n", iter->hastaNo);

printf("Ad: %s\n", iter->ad);

printf("Yaş: %d\n", iter->yas);

printf("Cinsiyet: %c\n", iter->cinsiyet);

printf("Mahkum: %d\n", iter->mahkum);

printf("Engellilik Oranı: %.2f\n", iter->engellilik);

printf("Kanamalı Hasta: %d\n", iter->kanamali);

printf("Kayıt Saati: %s\n", iter->kayitSaat);

printf("Muayene Saati: %s\n", iter->muayeneSaat);

printf("Muayene Süresi: %d dakika\n", iter->muayeneSure);

printf("Öncelik: %d\n", iter->oncelik);

printf("\n");

iter = iter->next;

}

}

int YasPuani(Hasta\* hasta) {

if (hasta->yas < 18)

return 5;

else if (hasta->yas >= 65)

return 3;

else

return 0;

}

int IsMahkum(Hasta\* hasta) {

return (hasta->mahkum == 1) ? 10 : 0;

}

int KanamaDegeri(Hasta\* hasta) {

return (hasta->kanamali == 1) ? 2 : 0;

}

int EngellilikPuani(Hasta\* hasta) {

float engellilikOrani = hasta->engellilik;

if (engellilikOrani >= 40.0)

return 7;

else if (engellilikOrani >= 20.0)

return 4;

else

return 0;

}

int OncelikPuaniHesapla(Hasta\* hasta) {

int oncelikPuan = 0;

oncelikPuan += YasPuani(hasta);

oncelikPuan += IsMahkum(hasta);

oncelikPuan += KanamaDegeri(hasta);

oncelikPuan += EngellilikPuani(hasta);

return oncelikPuan;

}

void HastaMuayeneEt() {

if (ilkHasta == NULL) {

printf("Hasta yok.\n");

return;

}

Hasta\* muayeneEdilen = ilkHasta;

ilkHasta = ilkHasta->next;

printf("Hasta Bilgileri:\n");

printf("Hasta No: %d\n", muayeneEdilen->hastaNo);

printf("Ad: %s\n", muayeneEdilen->ad);

printf("Yaş: %d\n", muayeneEdilen->yas);

printf("Cinsiyet: %c\n", muayeneEdilen->cinsiyet);

printf("Mahkum: %d\n", muayeneEdilen->mahkum);

printf("Engellilik Oranı: %.2f\n", muayeneEdilen->engellilik);

printf("Kanamalı Hasta: %d\n", muayeneEdilen->kanamali);

printf("Kayıt Saati: %s\n", muayeneEdilen->kayitSaat);

printf("Muayene Saati: %s\n", muayeneEdilen->muayeneSaat);

printf("Muayene Süresi: %d dakika\n", muayeneEdilen->muayeneSure);

printf("Öncelik: %d\n", muayeneEdilen->oncelik);

if (sonHasta == muayeneEdilen) {

sonHasta = NULL;

}

free(muayeneEdilen);

if (ilkHasta != NULL) {

Hasta\* enYuksekOncelikliHasta = ilkHasta;

Hasta\* iter = ilkHasta;

while (iter != NULL) {

if (iter->oncelik > enYuksekOncelikliHasta->oncelik) {

enYuksekOncelikliHasta = iter;

}

iter = iter->next;

}

int muayeneBitisSaat = saatleriTopla(enYuksekOncelikliHasta->kayitSaat, enYuksekOncelikliHasta->muayeneSaat);

char muayeneBitisSaatStr[6];

sprintf(muayeneBitisSaatStr, "%04d", muayeneBitisSaat);

printf("Bir sonraki hasta muayene giriş saati: %s\n", muayeneBitisSaatStr);

}

}

int main() {

HastalariYukle();

HastaNoEkle();

Yazdir();

HastaMuayeneEt();

return 0;

}

\*/