#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

typedef struct NO{

int nivel;

char descrisao[100];

struct NO \*prox;

}tno;

void inserir(tno \*\*L,char descri[]);

void Procurar(tno \*L,char des[]);

void executarTarefa(tno \*\*L,tno \*\*Execu);

void exibirNivel(tno \*L,int n);

void exibirTudo(tno \*L);

void alterarNivel(tno \*\*L,char descri[]);

void limpar(tno \*\*L);

tno \* buscar(tno \*L,char n[]);

int main(){

tno \*Lista=NULL,\*ListaExecu=NULL;

char op;

char aux[100];

int nivel;

do{

printf("MENU\n1-Inserir nova tarefa\n2-Procurar tarefa\n3-Executar tarefa\n"

"4-Cancelar Execucao da tarefa\n5-Exibir Decricao de todas tarefas de uma prioridade\n6-Exibir Todas atividades"

"\n7-Alterar prioridade da tarefa\n8-Limpar lista de tarefas\n9-Sair\nDigite sua opcao: ");

scanf("%c",&op);

switch (op)

{

case '1':

printf("digite a descrição da tarefa: ");

gets(aux);

inserir(&Lista,aux);

break;

case '2':

printf("digite a descrição da tarefa: ");

gets(aux);

break;

case '3':

break;

case '4':

break;

case '5':

printf("Digite o nivel da prioridade: ");

scanf("%i",&nivel);

exibirNivel(Lista,nivel);

break;

case '6':

exibirTudo(Lista);

break;

case '7':

printf("Digite a descricao: ");

gets(aux);

alterarNivel(&Lista,aux);

break;

case '8':

limpar(&Lista);

break;

case '9':

break;

default:

printf("Opcao invalida!\n");

break;

}

}while(op!='8');

return 0;

}

void inserir(tno \*\*L,char descri[]){

tno \*novo,\*teste,\*auxPont,\*auxPont2;

int aux;

teste = buscar((\*L),descri);

if(teste != NULL){

printf("tarefa existente! Operaçao cancelada!\n");

return;

}

else{

novo = (tno \*) malloc(sizeof(tno));

printf("Digite o nivel de prioridade da tarefa: ");

scanf("%i",&novo->nivel);

if((\*L)==NULL){//lista vazia

strcpy(novo->descrisao,descri);

novo->prox = NULL;

(\*L) = novo;

}

else{

if((\*L)->prox==NULL){// uma tarefa

if((\*L)->nivel > novo->nivel){

(\*L)->prox = novo;

novo ->prox = NULL;

}

else{

novo ->prox = \*L;

\*L = novo;

}

}

else{// mais de 1 tarefa

auxPont=(\*L);

auxPont2 = auxPont->prox;

while(auxPont2!=NULL){

if(auxPont->nivel < novo->nivel || auxPont->nivel == novo->nivel)

break;

if(auxPont2->nivel < novo->nivel || auxPont2->nivel == novo->nivel)

break;

auxPont=auxPont->prox;

auxPont2=auxPont2->prox;

}

if(auxPont2==NULL){//fim

auxPont->prox=novo;

novo->prox=NULL;

}

else{

auxPont->prox=novo;

novo->prox=auxPont2;

}

}

}

}

}

tno \*buscar(tno \*L,char n[]){

tno \*aux=L;

while(aux!=NULL){

if(strcmp(aux->descrisao,n)==0)

break;

aux = aux ->prox;

}

return aux;

}

void Procurar(tno \*L,char des[]){

tno \*aux=L;

int i=0;

while(aux!=NULL){

if(strcmp(aux->descrisao,des)==0)

break;

i++;

}

printf("Tarefa encontrada ! Exitem %i tarefas a serem executadas antes da procurada !",i+1);

}

void exibirTudo(tno \*L){

tno \*aux=L;

int i=0;

while(aux != NULL){

printf("Tarefa %i:\n-Descricao = %s\n-Prioridade=%i\n\n",i+1,aux->descrisao,aux->nivel);

i++;

}

printf("Voce tem um total de %i tarefas.",i+1);

}

void exibirNivel(tno \*L,int n){

tno \*aux;

int i, achou = 0,qtd=0;

for (aux = L; aux != NULL ; aux=aux->prox) {

if (aux->nivel == n) {

achou = 1;

printf("Tarefa %i:\n-Descricao = %s\n\n",i+1,aux->descrisao);

qtd++;

}

else if (aux->nivel < n)

break;

}

if(achou == 0){

printf("Nenhuma tarefa desde nivel de prioridade foi encontrada!\n");

return;

}

printf("O total de tarefas encontradas com este nivel de prioridade: %i",qtd+1);

}

void alterarNivel(tno \*\*L,char descri[]){

tno \*aux,\*auxPont,\*auxPont2,\*pos;

aux = (\*L);

pos = buscar(aux,descri);

if(pos == NULL){

printf("descricao nao existente!\n");

return;

}

else{

if(strcmp(aux->descrisao,descri)==0){

(\*L)= aux ->prox;

}

else{

while(aux!=NULL){

if(strcmp(aux->prox->descrisao,descri)==0)

break;

aux = aux->prox;

}

aux->prox=aux->prox->prox;

}

free(pos);

inserir(L,descri);

}

}

void limpar(tno \*\*L){

tno \*aux;

aux = (\*L)->prox;

while(\*L!=NULL){

free(\*L);

\*L=aux;

aux = aux->prox;

}

}