Questao unica)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct Ttarefa

{

char descricao[100];

int prioridade;

struct Ttarefa \*prox,\*ant;

}TContato;

void criarNO(TContato \*L,char n[]); // DOne !

void Adicionar(TContato \*\*a,char n[]); // Done !

void executarTarefa(TContato \*\*a);

void ProcurarTarefa(TContato \*L,char n[]);

void Exibir\_Lista(TContato \*a);

void Exibir\_Tarefas(TContato \*a,int n);

void CancelarTarefa(TContato \*\*a,char n[]);

void AlterarPrioridade(TContato\*\*a,char n[]);

void Remover\_Tudo(TContato \*\*a);

int main(){

TContato \*Lista=NULL;

int nun;

char op,aux[150];

do{

fflush(stdin);

printf("1-Inserir nova tarefa\n"

"2-Procurar tarefa\n"

"3-Executar tarefa\n"

"4-Cancelar tarefa\n"

"5-Exibir tarefas de uma prioridade\n"

"6-Exibir lista de tarefas\n"

"7-Alterar prioridade de uma tarefa\n"

"8-Sair\n"

"Digite sua opcao: ");

op = getchar();fflush(stdin);

switch(op){

case '1':

printf("Digite a descricao da tarefa: ");

gets(aux);

Adicionar(&Lista,aux);

break;

case '2':

printf("Digite a descricao da tarefa: ");

gets(aux);

ProcurarTarefa(Lista,aux);

break;

case '3':

executarTarefa(&Lista);

break;

case '4':

printf("Digite a descricao da tarefa: ");

gets(aux);

CancelarTarefa(&Lista,aux);

break;

case '5':

printf("Digite a prioridade das tarefas: ");

scanf("%d",&nun);

Exibir\_Tarefas(Lista,nun);

break;

case '6':

Exibir\_Lista(Lista);

break;

case '7':

printf("Digite a descricao da tarefa: ");

gets(aux);

AlterarPrioridade(&Lista,aux);

break;

case '8':

break;

default:

printf("\nOpcao invalida!\n");

break;

}

}while(op!='8');

Remover\_Tudo(&Lista);

return 0;

}

void criarNO(TContato \*\*L,char n[],int nun){

\*L = (TContato \*) malloc(sizeof(TContato));

strcpy((\*L)->descricao,n);fflush(stdin);

(\*L)->prioridade = nun;

}

void Adicionar(TContato \*\*L,char n[]){

TContato \*novo,\*aux,\*local=NULL;

int prioAux;

printf("digite a prioridade (1 ate 10) : ");

scanf("%d",&prioAux);

if(prioAux < 1 || prioAux > 10){

printf("\nPrioridade invalida! operacao cancelada!\n");

return;

}

if(\*L == NULL){

criarNO(&novo,n,prioAux);

novo->ant = NULL;

novo->prox = NULL;

\*L = novo;

}

else{

aux = \*L;

while(aux->prox != NULL){

if(strcmp(aux->descricao, n ) == 0){

printf("\nDescricao repetida! Operacao cancelada!\n\n");

return;

}

if(prioAux > aux->prioridade)

local = aux;

aux = aux->prox;

}

if(strcmp(aux->descricao, n ) == 0){

printf("\nDescricao repetida! Operacao cancelada!\n\n");

return;

}

if(prioAux > aux->prioridade)

local = aux;

if(local != NULL){//---------- Quando encontra o local

if(local == \*L){// inserir no inicio

criarNO(&novo,n,prioAux);

novo->ant = NULL;

novo->prox = local;

local->ant = novo;

\*L = novo;

}

else{// inserir no local

criarNO(&novo,n,prioAux);

novo->prox = local;

novo->ant = local ->ant;

local->ant->prox =novo;

local->ant = novo;

}

}

else{// inserir no fim

criarNO(&novo,n,prioAux);

novo->prox = NULL;

novo->ant = aux;

aux->prox = novo;

}

}

}

void ProcurarTarefa(TContato \*L,char n[]){

int qtd=0;

while(L!=NULL){

if(strcmp(L->descricao, n ) == 0){

printf("\n\nTarefa Encontrada!\nExistem %d tarefas a frente .\n\n");

return;

}

qtd++;

L = L->prox;

}

printf("\n\nNenhuma tarefa foi encontrada com essa descricao!\n\n");

}

void Exibir\_Tarefas(TContato \*L,int n){

int qtd=0;

if(L == NULL)

printf("Lista Vazia!\n\n");

else{

while(L!=NULL){

if(n > L->prioridade)

break;

if(L->prioridade == n){

printf("\n\nTarefa %d:\n Descricao: %s\n",qtd+1,L->descricao);

}

qtd++;

L = L->prox;

}

if(qtd == 0)

printf("\n\nNenhuma tarefa foi encontrada com essa prioridade!\n\n");

}

}

void executarTarefa(TContato \*\*L){

TContato \*aux;

if(\*L == NULL)

printf("Lista vazia!");

else{

aux = \*L;

\*L = (\*L)->prox;

(\*L)->ant = NULL;

free(aux);

}

}

void CancelarTarefa(TContato \*\*a,char n[]){

TContato \*aux;

if(\*a == NULL){

printf("Lista vazia!\n");

}

else if(strcmp((\*a)->descricao, n ) == 0){// se for o 1º

if((\*a)->prox == NULL){ // 1 unico no

aux = \*a;

\*a = NULL;

free(aux);

}

else{

aux = \*a;

\*a = (\*a)->prox;

free(aux);

}

}

else{ // mais de 1

aux = aux->prox;

do{

if(strcmp(aux->descricao, n ) == 0){

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

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

free(aux);

return;

}

aux = aux->prox;

}while(aux!=NULL);

printf("\n\nTarefa nao encontrada!\n\n");

}

}

void Exibir\_Lista(TContato \*a){

if(a == NULL)

printf("Lista vazia !\n\n");

else{

while(a!=NULL){

printf("Descricao da tarefa: %s\n"

"Prioridade: %d\n",a->descricao,a->prioridade);

a = a->prox;

}

}

}

void AlterarPrioridade(TContato\*\*a,char n[]){

TContato \*aux,\*local=NULL,\*no=NULL;

int prioAux;

printf("digite a prioridade (1 ate 10) : ");

scanf("%d",&prioAux);

if(prioAux < 1 || prioAux > 10){

printf("\nPrioridade invalida! operacao cancelada!\n");

return;

}

if(\*a==NULL)// vazia

printf("Lista Vaiza!\n\n");

else if((\*a)->prox==NULL){// 1 no

if(strcmp((\*a)->descricao, n ) == 0){

(\*a)->prioridade = prioAux;

}

else

printf("\nDescricao nao encontrada!\n\n");

}

else{// mais de 1 no

if(strcmp((\*a)->descricao, n ) == 0){// se for o 1º

no = \*a;// guarda o ende. do 1º no

if(prioAux >= (\*a)->prioridade){// se a prioridade nova do 1º for maior q a dele mesmo

(\*a)->prioridade = prioAux;

}

else if(prioAux >= (\*a)->prox->prioridade){// se a prioridade nova do 1º for maior q a do 2º

(\*a)->prioridade = prioAux;

}

else{// se for menor que o 2º, deve modifacar o "\*a"

no ->prioridade = prioAux;

aux = (\*a)->prox;

(\*a)->ant = NULL;

\*a = aux;

while(aux->prox!=NULL){

if(aux->prioridade <= prioAux){

local = aux;

break;

}

aux = aux->prox;

}

if(aux->prox != NULL ){ // meio

no ->prox = local;

no->ant = local->ant;

local->ant->prox = no;

local ->ant = no;

}

else{ // no fim

if(aux->prioridade < prioAux){//antes do fim

no ->prox = aux;

no->ant = aux->ant;

aux->ant->prox = no;

aux ->ant = no;

}

else{// depois do ultimo

aux ->prox = no;

no ->ant = aux;

no ->prox = NULL;

}

}

}

}

else{// ---- 2º em diante

aux = (\*a)->prox;

while(aux!=NULL){

if(strcmp(aux->descricao, n ) == 0){ // procurando o No que sera alterado

no = aux;

no->ant->prox = no->prox;

no->prioridade = prioAux;

if(aux->prox != NULL)

no->prox->ant = no->ant;

break;

}

aux = aux->prox;

}

if(no != NULL){

aux = \*a;

while(aux->prox!=NULL){ // --- procurando o local da insersao

if(aux->prioridade <= prioAux){

local = aux;

break;

}

aux = aux->prox;

}

if(aux->prox != NULL){

no ->prox = local;

no->ant = local->ant;

if(local != \*a){ // se o local for antes do 1º no

local->ant->prox = no;

}

else{

\*a = no; // novo inicio da lista

}

local ->ant = no;

}

else{ // no fim

aux ->prox = no;

no ->ant = aux;

no->prox = NULL;

}

}

else

printf("\nTarefa nao encontrada!\n\n");

}

}

}

void Remover\_Tudo(TContato \*\*a){

TContato \*aux;

if(\*a != NULL){

do{

aux = \*a;

\*a = (\*a)->prox;

free(aux);

}while(\*a != NULL);

}

}