1°)

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

typedef struct No{

int faltas;

float media;

char nome[100];

struct No \*prox;

}Tno;

void cadastrar(Tno \*\*L);

void listar(Tno \*L);

Tno \*consultar(Tno \*L,char nome[]);//busca

void exibir(Tno \*L,char nome[]);

void alterar\_Media(Tno \*L,char nome[]);

void alterar\_Falta(Tno \*L,char nome[]);

void remover(Tno \*\*L,char nome[]);

void limpar(Tno \*\*L);

int main(){

Tno \*lista,\*endAux;

char op,aux[100];

do{

system("cls");

printf("\sM E N U\nO que deseja realizar:"

"\n1-Cadastrar Aluno\n2-Listar\n3-Alterar media\n"

"4-Alterar Faltas\n5-Exibir Aluno\n6-Remover Aluno\n7-Limpar\n8-Sair\nDigite sua opcao: ");

scanf("%c",&op);fflush(stdin);

switch(op){

case'1':

cadastrar(&lista);

break;

case'2':

listar(lista);

break;

case'3':

printf("Digite o nome do aluno: ");

gets(aux);

alterar\_Falta(lista,aux);

break;

case'4':

printf("Digite o nome do aluno: ");

gets(aux);

alterar\_Media(lista,aux);

break;

case'5':

printf("Digite o nome do aluno: ");

gets(aux);

exibir(lista,aux);

break;

case'6':

printf("Digite o nome do aluno: ");

gets(aux);

remover(&lista,aux);

break;

case'7':

limpar(&lista);

break;

case'8':

break;

default:

printf("Opcao invalida!\n");

break;

}

system("pause");

}while(op != '8');

return 0;

}

Tno \*consultar(Tno \*L,char nome[]){

Tno \*aux=L;

if(L==NULL)

printf("Lista Vazia!\n");

else{

do{

if(strcmp(aux->nome,nome)==0)

return aux;

aux=aux->prox;

}while(aux!=L);

}

return NULL;

}

void cadastrar(Tno \*\*L){

Tno \*novo,\*aux;

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

printf("Digite o nome do aluno: ");

gets(novo->nome);

printf("Digite a media do aluno: ");

scanf("%f",&novo->media);

printf("Digite a quantidade de faltas do aluno: ");

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

novo->prox = NULL;

if(\*L==NULL){

\*L=novo;

novo ->prox = novo;

}

else{

novo->prox=\*L;

aux = \*L;

while(aux->prox!=\*L)

aux=aux->prox;

aux ->prox = novo;

novo ->prox = \*L;

}

}

void listar(Tno \*L){

Tno \*aux=L;

if(aux == NULL)

printf("Lista vazia!\n");

else{

printf("\nAluno: %s\nMedia: %f\nQtd.Faltas: %i");

aux = aux ->prox;

while(aux != L){

printf("\nAluno: %s\nMedia: %f\nQtd.Faltas: %i");

aux = aux ->prox;

}

}

}

void alterar\_Media(Tno \*L,char nome[]){

Tno \*aux;

aux = consultar(L,nome);

if(aux == NULL){

printf("Nome nao encontrado!\n");

return;

}

else{

printf("Digite a nova media do aluno %s: ",aux->nome);

scanf("%f",&(aux->media));

}

}

void alterar\_Falta(Tno \*L,char nome[]){

Tno \*aux;

int qtd;

char op;

aux = consultar(L,nome);

if(aux == NULL){

printf("Operacao nao pode ser efetuada, pois o nome nao foi encontrado!\n");

return;

}

else{

do{

printf("O que Deseja realizar:\n1-Acresentar\n2-Retirar\n"

"Digite a opcao: ");

scanf("%c",&op);

}while(op != '1' && op != '2');

if(op == '1'){

printf("Digite quanto deseja adicionar: ");

scanf("%i",&qtd);

aux ->faltas = aux->faltas+qtd;

}

else{

printf("Digite quanto deseja retirar: ");

scanf("%i",&qtd);

aux ->faltas = aux->faltas-qtd;

}

}

}

void exibir(Tno \*L,char nome[]){

Tno \*aux=L,\*aux2;

do{

if(strcmp(aux->nome,nome)==0){

printf("Aluno : %s\nMedia Final: %f\nQuantidade de Faltas: %i\n",aux->nome,aux->media,aux->faltas);

return;

}

aux = aux ->prox;

}while(aux!=L);

printf("Nome nao cadastrado!\n");

}

void remover(Tno \*\*L,char nome[]){

Tno \*aux,\*aux2,\*aux3;

aux = consultar(\*L,nome);

if(aux == NULL){

printf("Operacao nao pode ser efetuada, pois o nome nao foi encontrado!\n");

return;

}

else{

aux2 = \*L;

aux3 = aux2 ->prox;

while(aux2->prox!=aux){

aux2 = aux3;

aux3 = aux3->prox;

}

if(aux2==aux3){//um no so

\*L = NULL;

free(aux3);

}

else{

aux2->prox=aux3->prox;

free(aux3);

}

}

}

void limpar(Tno \*\*L){

Tno \*aux,\*proxAux;

aux = \*L;

proxAux = aux ->prox;

while(aux!=proxAux){

aux->prox=proxAux->prox;

free(proxAux);

}

\*L = NULL;

free(aux);

}

2°)

#include <stdio.h>

typedef struct Pessoa{

int info;

struct Pessoa \*prox;

}Tno;

int analizar(Tno \*L,Tno \*L2);

int analizar(Tno \*L,Tno \*L2){

Tno \*aux=L,\*aux2=L2;

int qtdDif=0;

do{

if(aux->info != aux2->info)

qtdDif++;

}while(aux!=L||aux2!=L2);

if(((aux==L && aux2!=L2)||(aux!=L&&aux2==L2)) || qtdDif > 0 )//tamanhos diferentes e quantidades de numeros diferentes

return 1;

else

return 0;

}

3°)

#include <stdio.h>

#include <stdlib.h>

typedef struct Pessoa{

int info;

struct Pessoa \*prox;

}Tno;

Tno \*Copiar(Tno \*L);

Tno \*Copiar(Tno \*L){

Tno \*aux=L,\*novo,\*Ante=NULL,\*primeiro;

do{

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

novo ->info = aux ->info;

if(Ante==NULL){

primeiro = novo;

Ante = novo;

novo ->prox = novo;

}

else{

Ante->prox=novo;

}

aux = aux ->prox;

}while(aux!=L);

novo ->prox=primeiro;

return primeiro;

}