Universidade Federal Fluminense

Matéria Estrutura de Dados

Professor Dalessandro Soares

Alunos: Alessandro Sampaio e Tatiane Sousa

#include <stdio.h>

#include <stdlib.h>

typedef struct aluno{

int matricula;

char nome[31];

}Aluno;

typedef struct professor{

int matricula;

char nome[31];

char titulacao[11];

char dependentes[51];

}Professor;

typedef struct lista{

int id;

char tipo;

void \*info;

struct lista \*prox;

}Lista;

Aluno\* novo\_aluno(){

Aluno \*no=(Aluno\*)malloc(sizeof(Aluno));

printf ("Digite a matricula:\n");

scanf ("%d",&no->matricula);

printf ("Digite o nome:\n");

scanf (" %30[^\n]",no->nome);

return no;

}

Professor\* novo\_professor(){

Professor \*no=(Professor\*)malloc(sizeof(Professor));

printf ("Digite a matricula:\n");

scanf ("%d",&no->matricula);

printf ("Digite o nome:\n");

scanf (" %30[^\n]",no->nome);

printf ("Digite a titulacao:\n");

scanf (" %11[^\n]",no->titulacao);

printf ("Digite os dependentes:\n");

scanf (" %50[^\n]",no->dependentes);

return no;

}

Lista\* inserir\_inicio(Lista \*l, void \*x, char tipo, int cont){

Lista \*no=(Lista\*)malloc(sizeof(Lista));

no->id=cont;

no->tipo=tipo;

no->info=x;

no->prox=l;

l=no;

return l;

}

Lista\* inserir\_fim(Lista \*l, void \*x, char tipo, int cont){

if (l==NULL)

return inserir\_inicio(l,x,tipo,cont);

else

l->prox=inserir\_fim(l->prox,x,tipo,cont);

}

int busca(Lista \*l, int mat){

Lista \*p=l;

while (p!=NULL){

if(p->tipo=='a' || p->tipo=='A'){

Aluno \*a=(Aluno\*)p->info;

if(a->matricula==mat)

return 1;

}

p=p->prox;

}

return 0;

}

int busca\_id(Lista \*l, int idi){

Lista \*p=l;

while(p!=NULL){

if(idi==p->id)

return 1;

p=p->prox;

}

return 0;

}

Lista\* remover\_aluno(Lista\*l, int mat){

Lista \*l1;

Aluno \*a1=(Aluno\*)l->info;

if(a1->matricula==mat){

l1=l;

l=l->prox;

free(l1->info);

free(l1);

}

else

l->prox=remover\_aluno(l->prox,mat);

return l;

}

Lista\* remove\_no(Lista \*l, int idi){

Lista \*l1;

if(l->id==idi){

l1=l;

l=l->prox;

free(l1->info);

free(l1);

}

else

l->prox=remove\_no(l->prox,idi);

return l;

}

void imprimir\_lista (Lista \*l){

Lista \*p=l;

while(p!= NULL){

if(p->tipo=='a' || p->tipo=='A'){

Aluno \*a=(Aluno\*)p->info;

printf ("Aluno\n");

printf ("ID: %d\n", p->id);

printf ("Matricula: %d\n", a->matricula);

printf ("Nome: %s\n", a->nome);

printf ("-----------------------------------\n");

}

else if (p->tipo=='p' || p->tipo=='P'){

Professor \*r=(Professor\*)p->info;

printf ("Professor\n");

printf ("ID: %d \n",p->id);

printf ("Matricula: %d \n",r->matricula);

printf ("Nome: %s\n",r->nome);

printf ("Titulacao: %s \n",r->titulacao);

printf ("Dependentes: %s \n",r->dependentes);

printf ("-----------------------------------\n");

}

p=p->prox;

}

}

int conta\_no(Lista \*l){

int cont=0;

Lista \*p=l;

while(p!=NULL){

cont++;

p=p->prox;

}

return cont;

}

Lista\* remover\_lista(Lista\* l){

if(l->prox==NULL){

free(l->info);

free(l);

}

else

l->prox=remover\_lista(l->prox);

return NULL;

}

int main (void){

Lista \*l=NULL;

Aluno \*a=NULL;

Professor \*p=NULL;

int op=0,cont=0,idi=0,y=0;

char tipo;

while(op!=8){

system ("cls");

printf ("[ - - Menu de opcoes - - ]\n\n");

printf ("1 - Inserir no inicio da lista\n");

printf ("2 - Inserir no fim da lista\n");

printf ("3 - Remover um determinado aluno da lista\n");

printf ("4 - Remover um noh da lista\n");

printf ("5 - Imprimir lista\n");

printf ("6 - Contar o numero de elementos na lista\n");

printf ("7 - Verificar se o elemento existe\n");

printf ("8 - Sair\n\n");

scanf ("%d",&op);

system ("cls");

switch (op){

case 1:{

printf("Digite a opcao para inserir no INICIO da lista:\n");

printf("(A) - Para inserir um aluno\n");

printf("(P) - Para inserir um professor\n");

scanf(" %c",&tipo);

printf("\n");

if(tipo=='a' || tipo=='A'){

a=novo\_aluno();

l=inserir\_inicio(l,a,tipo,cont);

}

else if(tipo=='p' || tipo=='P'){

p=novo\_professor();

l=inserir\_inicio(l,p,tipo,cont);

}

cont++;

system ("pause");

break;

}

case 2:{

printf("Digite a opcao para inserir no FIM da lista:\n");

printf("(A) - Para inserir um aluno\n");

printf("(P) - Para inserir um professor\n");

scanf(" %c",&tipo);

printf("\n");

if(tipo=='a' || tipo=='A'){

a=novo\_aluno();

l=inserir\_fim(l,a,tipo,cont);

}

else if(tipo=='p'||tipo=='P'){

p=novo\_professor();

l=inserir\_fim(l,p,tipo,cont);

}

cont++;

system ("pause");

break;

}

case 3:{

int mat=0;

printf ("Digite a matricula do aluno a ser removido\n");

scanf ("%d",&mat);

y=busca(l,mat);

if(y==1){

l=remover\_aluno(l,mat);

printf("Aluno removido com sucesso\n");

}

else

printf("Aluno nao consta na lista\n");

system ("pause");

break;

}

case 4:{

printf("Digite a ID do noh a ser removido\n");

scanf("%d",&idi);

y=busca\_id(l,idi);

if(y==1){

l=remove\_no(l,idi);

printf ("Noh %d removido com sucesso\n", idi);

}

else

printf ("Noh %d nao consta na lista\n", idi);

system ("pause");

break;

}

case 5:{

imprimir\_lista(l);

system ("pause");

break;

}

case 6:{

y=conta\_no(l);

printf("O numero de noh(s) eh/sao: %d\n\n",y);

system ("pause");

break;

}

case 7:{

printf("Digite a ID do noh a ser verificado\n");

scanf("%d",&idi);

y=busca\_id(l,idi);

if(y==1)

printf ("O noh %d consta na lista\n", idi);

else

printf ("O noh %d NAO consta na lista\n", idi);

system ("pause");

break;

}

case 8:{

l=remover\_lista(l);

system ("pause");

break;

}

}

}

return 0;

}