Listas:

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

typedef struct no No;

struct no{

int num;

struct no \*prox, \*search;

};

No\* criar\_no(){

No \*novo =(No\*)malloc(sizeof(No));

return novo;

}

//INSERIR NO INICIO

No\* inserir\_no\_inicio(No\* Lista,int dado){

No \*novo\_no=criar\_no();

novo\_no->num=dado;

if(Lista == NULL){

Lista=novo\_no;

novo\_no->prox=NULL;

}else{

novo\_no->prox=Lista;

Lista=novo\_no;

}

return Lista;

}

//INSERIR NO FINAL

No\* inserir\_no\_final(No\* Lista, int dado){

No \*novo\_no=criar\_no();

novo\_no->num=dado;

if(Lista==NULL){

novo\_no->prox=NULL;

Lista=novo\_no;

}else{

No\* aux=Lista;

while(aux->prox!=NULL){

aux=aux->prox;

}

novo\_no->prox=NULL;

aux->prox=novo\_no;

}

return Lista;

};

//INSERIR NO MEIO

No\* inserir\_no\_meio(No\* Lista, int dado){

No \*aux=Lista;

int i=0;

while(aux!=NULL){

aux=aux->prox;

i++;

}

//5

if (i < 0)

return;

if (i == 0)

{

inserir\_no\_inicio(Lista, dado);

return;

}

int p = i / 2;//2

if (i >= p)

{

No \*no = Lista, \*aux, \*novo;

novo = (No\*) malloc (sizeof(No));

novo->num = dado;

int cont = 0;//para contagem da posição a partir de 0

while (cont < p-1)//i-1 para parar uma posição antes de i, para fazer a inserção na posição certa

{

no = no->prox;

cont++;

}

aux = no->prox;

no->prox = novo;

novo->prox = aux;

}

return Lista;

};

//Contar

void contar\_nos(No\* Lista){

No \*aux=Lista;

int i=0;

while(aux!=NULL){

aux=aux->prox;

i++;

}

printf("Existem %d items na nossa lista.", i);

}

//IMPRIMIR LISTA

void imprimir\_lista(No\* Lista){

No \*aux=Lista;

while(aux!=NULL){

printf("%d ", aux->num);

aux=aux->prox;

}

}

//LISTA VAZIA

void lista\_vazia(No\* Lista){

if(Lista->prox==NULL){

printf("\nA lista está vazia. ");

}else{

printf("\nA lista não esta vazia. ");

}

}

//Primeiro nó

void mostra\_primeiro\_no(No\* Lista)

{

No \*aux=Lista;

if(aux->prox!=NULL){

printf("\n O primeiro Numero é: %d \n", aux->num);

}else{

printf("\n Lista esta vazia.");}

}

//Último nó

void mostra\_ultimo\_no(No\* Lista)

{

No \*aux=Lista;

while(aux->prox!=NULL){

aux=aux->prox;

}printf("O último numero é: %d ", aux->num);

}

//Buscar na lista

struct No\* lista\_buscar(No\* Lista, int dado){

No \*aux = Lista;

if(aux->num == dado)

return Lista;

if(aux->prox!=NULL){

//printf("Valor %d não foi encontrado...\n\n", dado);

return lista\_buscar(aux->prox, dado);

return NULL;

}

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

// return lista\_buscar(aux->prox, dado);

//return NULL;

}

No\* tamanho(No\* Lista) //Retona o numeros de nós da Lista

{

No \*aux=Lista;

int i=0;

while(aux!=NULL){

aux=aux->prox;

i++;

}

return i;

}

//Remover da lista

No\* removerPosicao(No\* Lista, int i)//Remove o nó da posição i

{

if (i < 0)

return;

if ((Lista != NULL) && (tamanho(Lista) >= i))

{

No \*no = Lista, \*aux, \*anterior;

int cont = 0;//para contagem da posição a partir de 0

while (cont < i-1)

{

anterior = no;

no = no->prox;

cont++;

}

aux = no->prox;

free(no);

anterior->prox = aux;

}

}

//MAIN

int main()

{

setlocale(LC\_ALL, "Portuguese");

No \*Lista=NULL;

int p;

printf("Digite que os numero deseja adicionar na sua lista.\n");

printf("{0} Para sair.\n");

while(1){

scanf("%d", &p);

if(p==0) break;

Lista=inserir\_no\_final(Lista, p);

}

while(1){

int m;

int n;

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\* MENU \*");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n(1) Inserir no inicio");

printf("\n(2) Inserir no final");

printf("\n(3) Inserir no meio");

printf("\n(4) Verificar Lista");

printf("\n(5) Mostra primeiro Item");

printf("\n(6) Mostra Ultimo item");

printf("\n(7) Remover item");

printf("\n(8) Contar nos");

printf("\n(9) Verificar item");

printf("\n(0) Sair");

printf("\n");

printf("\nDigite o que deseja fazer: ");

scanf("%d", &m);

if(m==0){ printf("Até mais!"); break;}

if(m==1){

printf("\n Você quer inserir no INICIO.");

printf("\n Qual numero deseja inserir?");

scanf("%d", &n);

if(n!=NULL){

Lista=inserir\_no\_inicio(Lista, n);

printf("Numero %d inserido ao inicio com sucesso.", n);

}else{

return;

}

}

if(m==2){

printf("\n Você quer inserir no FINAL.");

printf("\n Qual numero deseja inserir?.");

scanf("%d", &n);

if(n!=NULL){

Lista=inserir\_no\_final(Lista, n);

printf("Numero %d inserido ao final com sucesso.", n);

}else{

return;

}

}

if(m==3){

printf("\n Você quer inserir no MEIO.[N FUNCIONA]");

printf("\n Qual numero deseja inserir?");

scanf("%d", &n);

if(n!=NULL){

Lista=inserir\_no\_meio(Lista, n);

printf("Numero %d inserido ao meio com sucesso.", n);

}else{

return;

}

}

if(m==4){

printf("\n");

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

imprimir\_lista(Lista);

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

}

if(m==5){

mostra\_primeiro\_no(Lista);

}

if(m==6){

mostra\_ultimo\_no(Lista);

}

if(m==7){

printf("\nVocê deseja remover o nó de que posição? ");

scanf("%d", &n);

if(n!=0){

removerPosicao(Lista, n);

printf("\nRemovido com sucesso.");

}else{

printf("\nPosição não encontrada.");

return;

}

}

if(m==8){

contar\_nos(Lista);

}

if(m==9){

printf("\nQual numero você deseja buscar?");

scanf("%d", &n);

if(n!=NULL){

lista\_buscar(Lista, n);

if(lista\_buscar == NULL)

printf("Valor %d nao foi encontrado...\n\n", n);

else

printf("Valor %d encontrado...\n\n");

}

}

}

}