UFU

Nome: Antonio Carlos Neto

Curso: Ciência da Computação

Disciplina: GBC024

MAIN.C :

#include <stdio.h>

#include <stdlib.h>

typedef struct ListElmt\_

{

int data;

struct ListElmt\_ \*next;

} ListElmt;

typedef struct List\_

{

int size;

ListElmt \*head;

ListElmt \*tail;

} List;

void insere\_head(List \*H, int x)

{

if(H->head==NULL)

{

H->head = (ListElmt\*) malloc(sizeof(ListElmt));

H->head->data = x;

H->head->next = H->tail;

H->tail = NULL;

H->size = 1;

}else

if(H->size==1)

{

H->tail = (ListElmt\*) malloc(sizeof(ListElmt));

H->tail->data = H->head->data;

H->tail->next = NULL;

H->size = 2;

H->head->next = H->tail;

H->head->data = x;

}else

if(H->size >= 2)

{

ListElmt\* aux;

aux = H->head->next;

H->head->next = (ListElmt\*) malloc(sizeof(ListElmt));

H->head->next->data = H->head->data;

H->head->next->next = aux;

H->head->data = x;

H->size = H->size + 1;

}

}

List inverter\_lista(List\* H)

{

if(H->head==NULL)

{

return(\*H);

}else

if(H->size==1)

{

return(\*H);

}else

{

List L;

L.head = NULL;

L.tail = NULL;

L.size = 0;

int i;

List aux = \*H;

for(i=H->size;i>0;i--)

{

insere\_head(&L,aux.head->data);

aux.head = aux.head->next;

}

return (L);

}

}

List corte\_lista(List\* H,int i, int j)

{

int k;

if(i>=j)

{

printf("\nErro, i e j invalidos");

return(\*H);

}

if(H->size < i || H->size < j)

{

printf("\nErro, lista nao suporta o corte");

return(\*H);

}

List J;

J.head = NULL;

J.tail = NULL;

J.size = 0;

List aux2 = \*H;

for(k=1;k<=j;k++)

{

if(k>=i)

{

insere\_head(&J,aux2.head->data);

}

aux2.head = aux2.head->next;

}

J=inverter\_lista(&J);

return(J);

}

//void imprimir\_lista(List H)

//{

// int i;

// if(H.head==NULL)

// {

// printf("\nERRO, Lista Vazia");

// }

// else

// {

// for(i=0;i < H.size;i++)

// {

// printf("\nElemento %d: %d",(i+1),H.head->data);

// H.head = H.head->next;

//

// }

// }

//

//}

//int main()

//{

// List L,T,C;

// L.head = NULL;

// L.tail = NULL;

// L.size = 0;

// int x,i,j;

// char c;

// do

// {

// printf("\nDeseja adicionar um elemento na lista: (Y,N)\n");

// scanf("%c",&c);

// fflush(stdin);

// if(c=='Y')

// {

// printf("\nDigite um numero a ser inserido: \n");

// scanf("%d",&x);

// fflush(stdin);

// insere\_head(&L,x);

// }

// }while(c !='N');

// imprimir\_lista(L);

// printf("\n\nA inversa da sua lista eh:");

// T = inverter\_lista(&L);

// imprimir\_lista(T);

//

// printf("\n\nPara cortar a lista digite:\ni : ");

// scanf("%d",&i);

// fflush(stdin);

// printf("j : ");

// scanf("%d",&j);

// fflush(stdin);

// C = corte\_lista(&L,i,j);

// imprimir\_lista(C);

//

// return 0;

//}

int main()

{

//a

List L,T,C;

L.head = NULL;

L.tail = NULL;

L.size = 0;

//imprimir\_lista(L);

//b

insere\_head(&L,30);

insere\_head(&L,20);

insere\_head(&L,10);

//imprimir\_lista(L);

//c

L = inverter\_lista(&L);

//imprimir\_lista(L);

//d

L = corte\_lista(&L,2,3);

//imprimir\_lista(L);

return 0;

}