Nome: João Paulo de Oliveira 11611BCC046

9° Aula prática

Uberlândia

2016

1.Código fonte:

#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(List\* lista,int dat){

ListElmt \*novo\_no;

novo\_no = (ListElmt\*) malloc(sizeof(ListElmt));

novo\_no->data = dat;

novo\_no->next = lista->head;

lista->head = novo\_no;

if(lista->size==0) lista->tail = novo\_no;

lista->size++;

}

List inversa(List lista){

List inver;

inver.size = 0;

ListElmt \*no;

no = lista.head;

int i;

for (i=0;i<lista.size;i++){

insere(&inver,no->data);

no = no->next;

}

return inver;

}

List sub(List lista, int inicio, int fim){

if(inicio>fim||fim>lista.size||inicio<1){

printf("Parametros Incorreto\n");

exit(1);

}

List sub;

sub.size = 0;

int i;

ListElmt \*no;

no = lista.head;

for(i=1;i<=lista.size;i++){

if(i>=inicio)

insere(&sub,no->data);

no = no->next;

}

return sub;

}

int main(){

List lista,inv,subli;

lista.size = 0;

int i;

for(i=30;i>1;i-=10){

insere(&lista,i);

}

ListElmt\* no;

no = lista.head;

for(i=0;i<lista.size;i++){

printf("%d ",no->data);

no = no->next;

}

inv = inversa(lista);

subli = sub(lista,2,3);

puts("\n");

no = subli.head;

for(i=0;i<subli.size;i++){

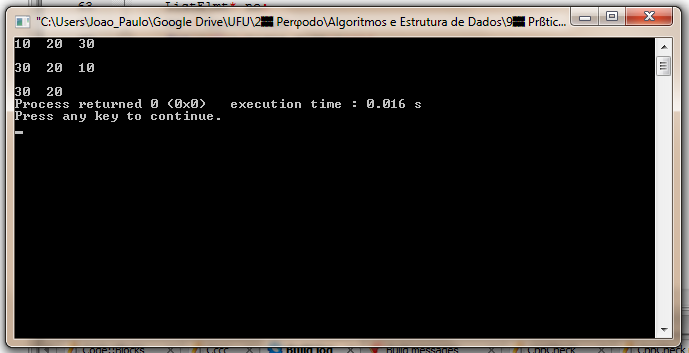
printf("%d ",no->data);

no = no->next;

}

return 0;

}

2.Print do funcionamento: