

**Nomes:**

**Henrique Sartori Siqueira**

**19240472**

**Rafael Silva Barbon**

**19243633**

**EX1:**

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
/*
```

Faça um programa que receba uma lista de valores inteiros e um valor inteiro n. Seu

programa deverá dividir a lista em duas, de tal forma que a segunda lista comece no primeiro

nó logo após a primeira ocorrência de n na lista original. Exibir os itens das duas listas após a

divisão. Caso n não exista na lista, exibir mensagem informativa.

```
*/
```

```
typedef struct lista{
```

```
    int info;
```

```
    struct lista *prox;
```

```
}Lista;
```

```
Lista * Inicializa(){
```

```
    return NULL;
```

```
}
```

```
Lista * Insere(Lista *recebida,int valor){
```

```
    Lista *novo = (Lista *)malloc(sizeof(Lista));
```

```
    novo->info = valor;
```

```
    novo->prox = recebida;
```

```
    return novo;
```

```
}
```

```
void Imprime(Lista *recebida){
```

```
    for(Lista *p = recebida; p!=NULL; p=p->prox){
```

```
        printf("Info = %d\n",p->info);
```

```
    }  
}
```

```
int Vazia(Lista *recebida){  
    if(recebida!=NULL)  
        return 0;  
    else  
        return 1;  
}
```

```
Lista * Busca(Lista *recebida,int v){  
    for(Lista *p = recebida; p!=NULL; p = p->prox){  
        if(p->info == v)  
            return p;  
    }  
    return NULL;  
}
```

```
void Libera(Lista *recebida){  
    Lista *aux = recebida, *aux2;  
    while(aux!=NULL){  
        aux2 = aux->prox;  
        free(aux);  
        aux = aux2;  
    }  
}
```

```
Lista * divide(Lista *l,int n){  
    Lista *aux = l, *l2 = Inicializa();  
    int x = 1;  
    while(aux != NULL){  
        if(aux->info == n){  
            l2 = aux->prox;  
            aux->prox = NULL;  
            x = 0;  
        }  
    }  
    return l2;  
}
```

```

        break;
    }
    aux = aux->prox;
}
if(x==1)
    printf("\n\tLista 2 nao pode ser criada.");
return l2;
}

```

```

int main(){
    Lista *l = Inicializa(), *l2 = Inicializa();
    int n;

    do{
        printf("\n\tInsira um valor para a lista (-111 para sair): ");
        scanf("%d",&n);
        if(n != -111)
            l = Insere(l,n);
    }while(n != -111);

    printf("\n\tInsira um valor para dividir a lista em duas: ");
    scanf("%d",&n);

    printf("\n\tLista original:\n");
    Imprime(l);

    l2 = divide(l,n);
    printf("\n\n\tLista 1:\n");
    Imprime(l);
    printf("\n\n\tLista 2:\n");
    Imprime(l2);

    Libera(l);
    Libera(l2);
    return 0;
}

```

```
}
```

## EX2:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
/*
```

Faça um programa que receba duas listas de inteiros L1 e L2. A quantidade de elementos em L1 e L2 pode ser diferente. Seu programa deverá verificar se a lista L1 está contida na lista L2 e exibir mensagem informativa. Exemplos: A lista L1 = {2,3} está contida na lista L2 = {1,2,3,4,5} A lista L1 = {1,2,3} não está contida na lista L2 = {4,5}

```
*/
```

```
struct lista{
```

```
    int info;
```

```
    struct lista *prox;
```

```
};
```

```
typedef struct lista Lista;
```

```
Lista* Inicializa(void){
```

```
    return NULL;
```

```
}
```

```
Lista* Insere(Lista *recebida, int valor){
```

```
    Lista *novo;
```

```
    novo = (Lista*)malloc(sizeof(Lista));
```

```
    novo->info=valor;
```

```
    novo->prox=recebida;
```

```
    return novo;
```

```
}
```

```
void Imprime(Lista *recebida){
```

```
    Lista *aux;
```

```
    for(aux=recebida; aux!=NULL; aux = aux->prox)
```

```
        printf(" %d", aux->info);
```

```
}
```

```
Lista* Busca(Lista *recebida,int valor)
{
    Lista *aux;
    for(aux=recebida;aux!=NULL;aux=aux->prox)
        if(aux->info==valor)
            return aux;
    return NULL;
}
```

```
Lista* Libera(Lista *L){
    Lista *aux;
    while(L!=NULL){
        aux=L->prox;
        free(L);
        L=aux;
    }
    return NULL;
}
```

```
int Vazia(Lista *recebida)
{
    if(recebida==NULL)
        return 1;
    return 0;
}
```

```
int Contida(Lista *L1, Lista *L2, int tam1)
{
    int infoaux, cont = 0;
    Lista *aux1;
    for(aux1=L1; aux1 != NULL; aux1=aux1 -> prox){
        infoaux = aux1->info;
        if(Busca(L2,infoaux)!=NULL){//Achou
```

```

        cont++;
    }
}
if(cont==tam1)//existem todos os valores de L1 em L2
    return 1;
return 0;
}

```

```

int main()
{
    Lista *L1=Inicializa(), *L2=Inicializa();
    int info, contL1=0;
    do{
        printf("\nInsira Valores para a lista 1(-999 p/ sair:");
        scanf("%d",&info);
        if(info!=-999){
            L1=Insere(L1, info);
            contL1++;
        }
    }while(info!=-999);
    system("clear");
    do{
        printf("\nInsira Valores para a lista 2:(-999 p/ sair)");
        scanf("%d",&info);
        if(info!=-999)
            L2=Insere(L2, info);
    }while(info!=-999);

    system("clear");
    printf("\nListas Inseridas:\nLista 1:\n\t{");
    Imprime(L1);
    printf(" }");
    printf("\nLista 2:\n\t{");
    Imprime(L2);
    printf(" }");
}

```

```
if(Contida(L1, L2, contL1))
    printf("\n\nA lista L1 esta contida em L2\n");
else
    printf("\n\nA lista L1 nao esta contida em L2\n");

L1=Libera(L1);
L2=Libera(L2);
if(Vazia(L1)&&Vazia(L2));
    printf("\nEncerrado com sucesso!!\n");
return 0;

}
```