

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Celula
{
    int conteudo;
    Celula *prox;
};

typedef Celula Lista;

Lista *crialista()
{
    Lista *lst;
    lst = NULL;

    return lst;
}

Celula *criaCelula(int cont)
{
    Celula *celula = (Celula *)malloc(sizeof(Celula));
    celula->conteudo = cont;
    celula->prox = NULL;

    return celula;
}

Lista *separa(Lista *l, int n)
{
    Celula *c;
    Lista *l2 = NULL;

    for (c = l; c != NULL; c = c->prox)
    {
        if (c->conteudo == n)
        {
            l2 = c->prox;
            c->prox = NULL;
            break;
        }
    }

    return l2;
}

int main()
{
    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa funcao separa_lista_encadeada>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    Lista *lista = criaLista();

    Celula *a = criaCelula(3);
    lista = a;

    Celula *b = criaCelula(17);
    a->prox = b;

    Celula *c = criaCelula(5);
    b->prox = c;

    Celula *d = criaCelula(12);
    c->prox = d;

    Celula *e = criaCelula(1);
```

```
d->prox = e;

printf("\n\nLISTA\n\n");

for (Celula *p = lista; p != NULL; p = p->prox)
{
    printf("%d ", p->conteudo);
}

printf("\n\nSeparando...\n\n");

Lista *lista2 = separa(lista, 5);

printf("\n\nLISTA\n\n");

for (Celula *p = lista; p != NULL; p = p->prox)
{
    printf("%d ", p->conteudo);
}

printf("\n\nLISTA 2\n\n");

for (Celula *p = lista2; p != NULL; p = p->prox)
{
    printf("%d ", p->conteudo);
}

//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
printf("\n\n\n");
system("pause");

//Retorno do método main
return(EXIT_SUCCESS);
}
```

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

struct lista
{
    float info;
    struct lista *prox;
};

typedef struct lista Lista;

Lista *concatena(Lista *l1, Lista *l2)
{
    if (l1 != NULL){
        lista *p = l1;

        while (p->prox != NULL)
        {
            p = p->prox;
        }

        p->prox = l2;
    }
    else
    {
        l1 = l2;
    }

    return l1;
}

lista *criaCelulaL(float info)
{
    lista *celula = (lista *)malloc(sizeof(lista));
    celula->info = info;
    celula->prox = NULL;

    return celula;
}

int main()
{
    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa funcao concatena_lista_encadeada>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    Lista *lista1 = NULL;
    Lista *lista2 = NULL;

    lista *a = criaCelulaL(2.1);
    lista *b = criaCelulaL(4.5);
    lista *c = criaCelulaL(1.0);

    lista *d = criaCelulaL(7.2);
    lista *e = criaCelulaL(9.8);

    lista1 = a;
    a->prox = b;
    b->prox = c;

    lista2 = d;
    d->prox = e;

    printf("\n\nLISTA 1\n\n");

    for (lista *p = lista1; p != NULL; p = p->prox)
```

```
{
    printf("%.1f ", p->info);
}

printf("\n\nLISTA 2\n\n");

for (lista *p = lista2; p != NULL; p = p->prox)
{
    printf("%.1f ", p->info);
}

printf("\n\nConcatenando...\n\n");

Lista *lista3 = concatena(lista1, lista2);

printf("\n\nLISTA 1\n\n");

for (lista *p = lista1; p != NULL; p = p->prox)
{
    printf("%.1f ", p->info);
}

printf("\n\nLISTA 2\n\n");

for (lista *p = lista2; p != NULL; p = p->prox)
{
    printf("%.1f ", p->info);
}

printf("\n\nLISTA 3 (Concatenada)\n\n");

for (lista *p = lista3; p != NULL; p = p->prox)
{
    printf("%.1f ", p->info);
}

//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
printf("\n\n\n");
system("pause");

//Retorno do método main
return(EXIT_SUCCESS);
}
```

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

struct list
{
    int info;
    struct list *prox;
};

typedef struct list Lista;

Lista *constroi(int n, int *v)
{
    Lista * lista = NULL;
    list *celulaAnt = NULL;

    for (int i = 0; i < n; i++)
    {
        list *celula = (list *)malloc(sizeof(list));
        celula->info = v[i];
        celula->prox = NULL;

        if (i == 0)
        {
            lista = celula;
        }
        else
        {
            celulaAnt->prox = celula;
        }

        celulaAnt = celula;
    }

    return lista;
}

int main()
{
    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa funcao constroi lista encad.>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    int vetor[5] = { 3, 8, 1, 7, 2 };

    printf("\n\nVETOR\n\n");

    for (int i = 0; i < 5; i++)
    {
        printf("%d ", vetor[i]);
    }

    Lista *lista = constroi(5, vetor);

    printf("\n\nLISTA\n\n");

    for (list *p = lista; p != NULL; p = p->prox)
    {
        printf("%d ", p->info);
    }

    //Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
    printf("\n\n\n");
    system("pause");

    //Retorno do método main
    return(EXIT_SUCCESS);
}
```

```
}
```

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

struct lst
{
    float info;
    struct lst *prox;
};

typedef struct lst Lista;

lst *criaCelulaLst(float info)
{
    lst *celula = (lst *)malloc(sizeof(lst));
    celula->info = info;
    celula->prox = NULL;

    return celula;
}

Lista *retira_prefixo(Lista *l, int n)
{
    Lista *lr = NULL;

    int cont = 0;

    for (lst *p = l; p != NULL; p = p->prox)
    {
        if (cont == n)
        {
            lr = p;
        }

        cont++;
    }

    return lr;
}

int main()
{
    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa remove n elementos da list. encad.>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    Lista *lista = NULL;

    lst *a = criaCelulaLst(2.1);
    lst *b = criaCelulaLst(4.5);
    lst *c = criaCelulaLst(1.0);
    lst *d = criaCelulaLst(7.2);
    lst *e = criaCelulaLst(9.8);

    lista = a;
    a->prox = b;
    b->prox = c;
    c->prox = d;
    d->prox = e;

    printf("\n\nLISTA\n\n");

    for (lst *p = lista; p != NULL; p = p->prox)
    {
        printf("%.1f ", p->info);
    }

    int numero;
```

```
printf("\n\nInforme o numero de elementos que deseja remover: ");
scanf("%d", &numero);

Lista *listaResultante = retira_prefixo(lista, numero);

printf("\n\nLISTA RESULTANTE\n\n");

for (lst *p = listaResultante; p != NULL; p = p->prox)
{
    printf("%.1f ", p->info);
}

//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
printf("\n\n\n");
system("pause");

//Retorno do método main
return(EXIT_SUCCESS);
}
```