

```
int recup (LISTA_DUP_ENC l, int k)
{
    if (k < 1 || k > tam(l))
    {
        printf ("\nERRO! Consulta invalida.\n");
        exit (3);
    }
    for (;k>1;l=l->prox,k--) ;
    return (l->inf);
}
```

Listas Duplamente Encadeadas

Com base no que foi visto implemente a operação `ret()` que compõem o TAD `LISTA_DUP_ENC`.

```
typedef struct nodo
{
    int inf;
    struct nodo * ant;
    struct nodo * prox;
}NODO;
typedef NODO * LISTA_DUP_ENC;
void cria_lista (LISTA_DUP_ENC *);
int eh_vazia (LISTA_DUP_ENC);
int tam (LISTA_DUP_ENC);
void ins (LISTA_DUP_ENC *, int, int);
int recup (LISTA_DUP_ENC, int);
void ret (LISTA_DUP_ENC *, int);
```

```
void ret (LISTA_DUP_ENC *pl, int k)
{
    NODO *aux;
    if (k < 1 || k > tam(*pl))
    {
        printf ("\nERRO! Posição invalida para
        retirada.\n");
        exit (4);
    }
    if (k==1) /*situações um e dois*/
    {
        aux = *pl;
        *pl = aux->prox;
```

```

if (*pl) /*situação dois*/
    (*pl)->ant=NULL;
    free (aux);
}
else /*situações três e quatro*/
{
    for (aux=(*pl)->prox; k>2; k--, aux=aux->prox);
    aux->ant->prox = aux->prox;
    if (aux->prox) /*situação quatro*/
        aux->prox->ant = aux->ant;
    free (aux);
}
}

```

Listas Duplamente Encadeadas – Exercício

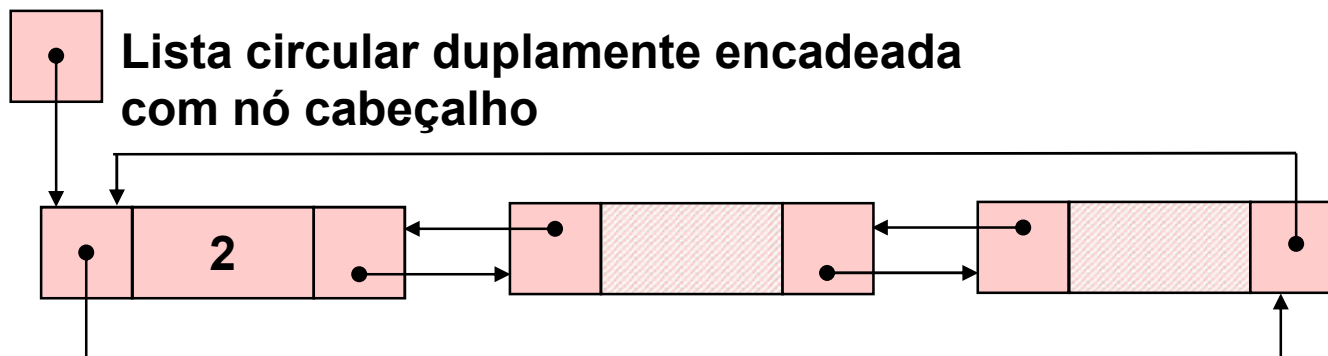
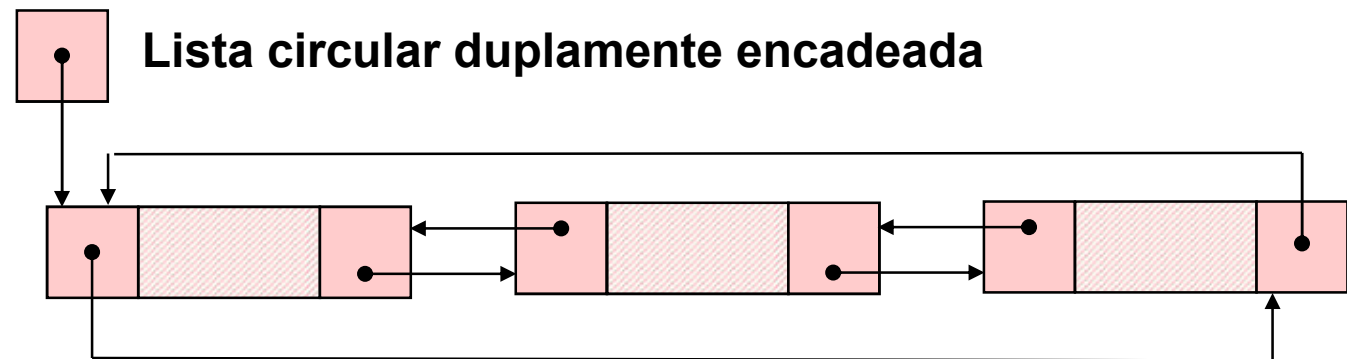
Implemente, no TAD LISTA_DUP_ENC, a seguinte operação:

```
void inverter_lista (LISTA_DUP_ENC *pl);
```

a qual recebe uma referência para uma lista duplamente encadeada e inverte a ordem de seus elementos.

Listas Duplamente Encadeadas

Também podemos construir *listas circulares duplamente encadeadas* ou *listas circulares duplamente encadeadas com nó cabeçalho*.



Listas Duplamente Encadeadas

/* Para uma melhor compreensão observe a definição do TAD LISTA_CIR_DUP_ENC_NC baixo: */

```
typedef struct nodo
```

```
{
```

```
    int inf;
```

```
    struct nodo * ant;
```

```
    struct nodo * prox;
```

```
}NODO;
```

```
typedef NODO * LISTA_CIR_DUP_ENC_NC;
```

```
void cria_lista (LISTA_CIR_DUP_ENC_NC *);
```

```
int eh_vazia (LISTA_CIR_DUP_ENC_NC);
```

```
int tam (LISTA_CIR_DUP_ENC_NC);
```

```
void ins (LISTA_CIR_DUP_ENC_NC, int, int);
```

```
int recup (LISTA_CIR_DUP_ENC_NC, int);
```

```
void ret (LISTA_CIR_DUP_ENC_NC, int);
```

Listas Duplamente Encadeadas

Com base no que foi visto implemente as operações do TAD LISTA_CIR_DUP_ENC_NC.

```
typedef struct nodo
{
    int inf;
    struct nodo * ant;
    struct nodo * prox;
}NODO;
typedef NODO * LISTA_CIR_DUP_ENC_NC;
void cria_lista (LISTA_CIR_DUP_ENC_NC *);
int eh_vazia (LISTA_CIR_DUP_ENC_NC);
int tam (LISTA_CIR_DUP_ENC_NC);
void ins (LISTA_CIR_DUP_ENC_NC, int, int);
int recup (LISTA_CIR_DUP_ENC_NC, int);
void ret (LISTA_CIR_DUP_ENC_NC, int);
```