```
/*
 * Faça uma biblioteca para as definições
 * abaixo especificadas.
 */

typedef int TipoChave;
typedef int TipoValor;
struct TipoListaEncadeada
{
  TipoChave chave;
  TipoValor valorQualquer;
  struct TipoListaEncadeada *prox;
  struct TipoListaEncadeada *ant;

};

typedef struct TipoListaEncadeada TipoListaEncadeada;

/*==================>PROCEDIMENTOS BÁSICOS DE LISTAS DUPLAMENTE ENCADEADA
/* -------------------------> Insercao inicio
 * Insere novo nó no início de uma lista duplamente encadeada.
 * A referência de ponteiro para o primeiro nó e os
 * valores dos campos do nó são dados.
 * Devolve endereço do nó recém inserido
 * ou NULL em caso de insucesso.
 */

TipoListaEncadeada *insereInicioListaEncadeada(TipoListaEncadeada **prim,
TipoChave chave, TipoValor valor);

/* -------------------------> Insercao final
 * Insere novo nó no final de uma lista duplamente encadeada.
 * A referência de ponteiro para o primeiro nó e os
 * valores dos campos do nó são dados.
 * Devolve endereço do nó recém inserido
 * ou NULL em caso de insucesso.
 */

TipoListaEncadeada *insereFimListaEncadeada(TipoListaEncadeada **prim,
TipoChave chave, TipoValor valor);


/* -------------------------> Remove nó por valor de chave
 * Remove nó cujo valor chave seja igual a 'chave'
 * Mantêm lista inalterada caso este não exista.
 */

void removeNo(TipoListaEncadeada **prim, TipoChave chave);

/* -------------------------> Cria cópia
 * Cria uma nova lista cujos nós são múltiplos de 3.
 * Devolve o ponteiro para
 * o primeiro nó da nova lista.
 */

TipoListaEncadeada *copiaLista3(TipoListaEncadeada *prim);
```