2.3/LSE.h

```c
#ifndef LISTASE_H
#define LISTASE_H

#include <stdio.h>
#include <stdlib.h>

typedef struct NO{
    int info;
    struct NO* prox;
}NO;

typedef struct NO* Lista;

Lista* criaLista(){
    Lista *li;
    li = (Lista*) malloc (sizeof(Lista));
    if(li != NULL){
        *li = NULL;
    }
    return li;
}

int listaVazia(Lista *li){
    if(li == NULL) return 1;
    if(*li == NULL) return 1;//True - Vazia!
    return 0;//False - tem elemento!
}

NO* alocarNO(){
    return (NO*) malloc (sizeof(NO));
}

void liberarNO(NO* q){
    free(q);
}

int listaBuscaElem(Lista* li, int elem, int *p){
    if(li == NULL) return 0;
    NO* aux = *li;
    while(aux != NULL){
        if(aux->info == elem){
            *p = aux->info;
            return 1;
        }
        aux = aux->prox;
    }
    return 0;
}

int insereIni(Lista* li, int elem){
    if(li == NULL) return 0;
    NO* novo = alocarNO();
    if(novo == NULL) return 0;
    novo->info = elem;
    novo->prox = *li;
    *li = novo;
    return 1;
```

```
58  }
59
60  int insereFim(Lista* li, int elem){
61      if(li == NULL) return 0;
62      NO* novo = alocarNO();
63      if(novo == NULL) return 0;
64      novo->info = elem;
65      novo->prox = NULL;
66      if(listaVazia(li)){
67          *li = novo;
68      }else{
69          NO* aux = *li;
70          while(aux->prox != NULL)
71              aux = aux->prox;
72          aux->prox = novo;
73      }
74      return 1;
75  }
76
77  int removeIni(Lista* li){
78      if(li == NULL) return 0;
79      if(listaVazia(li)) return 0;
80      NO* aux = *li;
81      *li = aux->prox;
82      liberarNO(aux);
83      return 1;
84  }
85
86  int removeFim(Lista* li){
87      if(li == NULL) return 0;
88      if(listaVazia(li)) return 0;
89      NO* ant, *aux = *li;
90      while(aux->prox != NULL){
91          ant = aux;
92          aux = aux->prox;
93      }
94      if(aux == *li)
95          *li = aux->prox;
96      else
97          ant->prox = aux->prox;
98      liberarNO(aux);
99      return 1;
100 }
101
102 void imprimeLista(Lista* li){
103     if(li == NULL) return;
104     if(listaVazia(li)){
105         printf("Lista Vazia!\n");
106         return;
107     }
108     //printf("Elementos:\n");
109     NO* aux = *li;
110     while(aux != NULL){
111         printf("%d ", aux->info);
112         aux = aux->prox;
113     }
114     printf("\n");
115 }
116
117 void destroiLista(Lista* li){
```

```
118     if(li != NULL){
119         NO* aux;
120         while((*li) != NULL){
121             aux = *li;
122             *li = (*li)->prox;
123             liberarNO(aux);
124         }
125         free(li);
126     }
127 }
128 
129 #endif
```