

## Lista para Verificação de Aprendizagem 2

Rodrigo de Souza

Dezembro de 2015

1. Descreva, em linguagem C, uma estrutura para implementação de listas duplamente ligadas circulares com cabeça para armazenamento de números inteiros. Escreva, também em linguagem C, as funções **insere** e **remove**, que respectivamente inserem um inteiro na lista, e que removem um inteiro da lista. Finalmente, escreva uma função **buscaRemove** que recebe a cabeça da lista e um inteiro, percorre a lista e, caso encontre uma célula cujo conteúdo seja esse inteiro, remove-a da lista. Escreva um comentário em cada função dizendo o que ela recebe e o que faz.
2. Critique a seguinte função, que promete remover um elemento de uma lista ligada circular com cabeça (o parâmetro recebe a cabeça da lista):

```
void remove (celula *p) {  
    celula *morta;  
    morta = p->prox;  
    if (morta->prox == NULL) p->prox = NULL;  
    else p->prox = morta->prox;  
    free (morta); }
```

3. Implemente uma fila de inteiros em uma lista encadeada simples sem célula-cabeça. Será preciso manter um ponteiro **ini** para a primeira célula e um ponteiro **fim** para a última. Exiba (sempre em C) a estrutura representando a célula da lista, e as funções **insere** e **remove**.
4. Escreva um algoritmo que use uma pilha para inverter a ordem das letras de cada palavra de uma string, preservando a ordem das palavras. Por exemplo, dado o texto ESTE EXERCICIO E MUITO FACIL a saída deve ser ETSE OICICREXE E OTIUM LICAF. Apresente seu resultado como uma função em linguagem C, com um comentário dizendo o que a função recebe, o que faz, e o que devolve.
5. Reescreva a função **infixaParaPosfixa**<sup>1</sup> supondo que a expressão pode ter colchetes além de parênteses.
6. Use o algoritmo **infixaParaPosfixa** (cf. questão anterior) para converter a expressão infixa  $(A+B*(C-D*(E-F)-G*H)-I*3)$  para a notação posfixa.

---

<sup>1</sup><http://www.ime.usp.br/~pf/algoritmos/aulas/pilha.html>

Elabore para isso uma tabela com três colunas, indicando, respectivamente, prefixo da expressão já lido, conteúdo da pilha, e expressão gerada, para cada iteração do algoritmo (um exemplo dessa tabela pode ser encontrado após a descrição da função).

7. Escreva uma função que recebe a raiz  $r$  de uma árvore binária de inteiros e um inteiro  $k$ , e apaga (libera da memória) todas as folhas da árvore contendo  $k$ . Pode supor que a árvore tem pelo menos dois nós. A estrutura resultante deve ser uma árvore legítima, sem ponteiros perdidos. Use recursão.