

Exercícios de Estruturas de Dados – Listas, pilhas, filas, deque

1. Escreva o procedimento `esvazia_pilha()`. Considere que as operações de manipulação da pilha (`empilha(item)`, `desempilha`, e `pilha_vazia()`) já existem e podem ser utilizados para escrever o `esvazia_pilha()`.
2. Sendo F uma fila circular de 4 elementos, qual a situação final da fila após a seguinte sequência de operações de inserção (I,valor) e remoção (R): I,3 - I,5 - I,7 - R - I,1 - R - I,4 - I,3 - R - I,1 - R ?
3. Qual a situação final das deque a seguir, após a execução das seguintes sequências de operações:
 - a) deque de entrada restrita: E-> A B
I(5), R(A), I(6), I(8), R(B), I(3), R(A), R(B)
 - b) deque de saída restrita: A B ->S
I(5,A), I(3,B), I(5,B), R, I(1,B), R, R, I(8,B), R
 - c) deque geral: E-> A B ->S
S<- <-E
I(5,A), I(3,B), I(5,B), R(A), I(1,B), R(B), R(A), I(8,B), R(B)
4. Um problema comum para compiladores e editores de texto é determinar se os parênteses em um string estão balanceados e aninhados corretamente. Por exemplo, na string `")()("` os parênteses não estão aninhados corretamente; e em `"())"` eles não estão balanceados. Considere que o string é implementado como uma lista de caracteres implementada por encadeamento simples. Escreva uma operação que retorna "verdadeiro" se o string contém parênteses balanceados, senão retorna "falso". *Dica: em nenhum momento serão encontrados mais fecha-parênteses do que abre-parênteses; use uma pilha para manter o número de parênteses encontrados.*

5. Implemente um interpretador de expressões, que tem por base a operação `calcula(...)`, a qual através uma pilha calcula o resultado de uma expressão aritmética. O procedimento `calcula(...)` recebe através de parâmetro uma pilha **p1** que contém números inteiros (os números da expressão aritmética). O procedimento deve ler um conjunto de caracteres (através de uma operação `le_caracter()` já existente), um caracter por vez. Se o caracter lido for “+” ou “-” ou “/” ou “*”, então o procedimento deve desempilhar dois números da pilha (se houver), efetuar a operação indicada e empilhar o resultado no topo da pilha. O procedimento termina quando existir apenas um número na pilha, que é o resultado da operação. Neste caso escrever o conteúdo do último nodo da pilha.

Outra opção é implementar o interpretador de expressões utilizando como entrada a expressão aritmética completamente parentizada, e utilizar duas pilhas, a pilha de operandos e a pilha de operadores (conforme exemplo apresentado nas notas de aula de pilhas, filas e deque, e utilizado em aula).

6. Dada uma lista L1, na qual cada nodo armazena um número inteiro, implementada com encadeamento simples, que pode conter números repetidos. Escreva operações (funções, métodos) em C/Java/C++/C# que:
- a) verifique se L1 está ordenada ou não (a ordem pode ser crescente ou decrescente);
 - b) faça uma cópia da lista L1 em uma outra lista L2;
 - c) faça uma cópia da Lista L1 em uma outra lista L2, eliminando elementos repetidos;
 - d) inverta L1 colocando o resultado em L2;
 - e) inverta L1 colocando o resultado na própria L1;
 - f) gere uma lista L3 que representa a concatenação de L1 e uma outra lista L2;
 - g) gere uma lista L2 onde cada registro contém dois campos de informação: *elem* contém um elemento de L1, e *count* contém quantas vezes este elemento apareceu em L1;
 - h) assumindo que os elementos da lista L1 são inteiros positivos, forneça os elementos que aparecem o maior e o menor número de vezes (forneça os elementos e o número de vezes correspondente);
7. Considere uma deque que manipula números inteiros cujo TAD define, entre outras, as seguintes operações: *InserInicio(x)*, *InserFim(x)*, *RemoveInicio()*, *RemoveFim()*, *ConsultaInicio()*, *ConsultaFim()*. Implemente uma deque de entrada restrita, utilizando lista duplamente encadeada, na qual a inserção so pode ser feita em uma das extremidades, e a remoção em ambas. Considere que as operações sobre pilhas e filas já existem, apenas faça chamadas a elas na sua implementação.
8. Como você implementaria uma lista de pilhas? Pense na estrutura de implementação (utilizando vetores? utilizando listas encadeadas? misturando as duas estruturas? como seriam as classes, funções, etc?), e implemente a operação *empilha(pilha, valor)* considerando a estrutura de implementação que você pensou.