



Lista de Exercícios sobre Pilhas

- 1) Use as operações/funções *Empilha* (*push*), *Desempilha* (*pop*) e *Vazia* (*Empty*) para construir operações que façam o seguinte:
 - a. Definir o item i como o segundo elemento a partir do topo da pilha, deixando a pilha sem seus dois elementos superiores (obs: no final, o item i será o topo da pilha).
 - b. Definir o item i como o segundo elemento a partir do topo da pilha, deixando a pilha inalterada (obs: alterar o segundo elemento).
 - c. Dado um inteiro n , definir o item i como o n -ésimo elemento a partir do topo da pilha, deixando a pilha sem seus n elementos superiores (obs: no final, o item i será o topo da pilha).
 - d. Dado um inteiro n , definir o item i como o n -ésimo elemento a partir do topo da pilha, deixando a pilha inalterada.
 - e. Definir o item i como o último elemento da pilha, deixando a pilha vazia (ou somente com o item i).
 - f. Definir o item i como o último elemento da pilha, deixando a pilha inalterada.
 - g. Definir o item i como o terceiro elemento a partir do final da pilha.

- 2) Escreva uma função para determinar se uma cadeia de caracteres (*string*) é da forma:
$$x C y$$
onde x e y são cadeias de caracteres compostas por letras ‘A’ e/ou ‘B’, e y é o inverso de x . Isto é, se $x = \text{“ABABBA”}$, y deve equivaler a “ABBABA” . Em cada ponto, você só poderá ler o próximo caractere da cadeia (é mandatório o uso de pilha).

- 3) Escreva uma função para determinar se uma cadeia de caracteres (*string*) é da forma:
$$a D b D c D \dots D z$$
onde cada cadeia de caracteres, a , b , ..., z , é da forma do exercício descrito acima. Portanto, uma cadeia de caracteres estará no formato correto se consistir em qualquer número de cadeias deste tipo ($x C y$), separadas pelo caractere ‘D’. Em cada ponto, você só poderá ler o próximo caractere da cadeia (é mandatório o uso de pilha).

- 4) Elabore um programa e as TADs necessárias, sem usar pilha, que simule uma sequência de operações *Empilha* (*push*) e *Desempilha* (*pop*), e determine se está ocorrendo um *underflow* (tentativa de Desempilhar em uma pilha vazia) ou não em alguma operação *Desempilha*.

- 5) Que conjunto de critérios é necessário e suficiente para que uma sequência de operações *Empilha* e *Desempilha* sobre uma única pilha (inicialmente vazia) deixe a pilha vazia e não provoque *underflow* (tentativa de Desempilhar em uma pilha vazia)? Que conjunto de critérios é necessário para que essa sequência deixe uma pilha não-vazia inalterada?



- 6) Para um dado número inteiro $n > 1$, o menor inteiro $d > 1$ que divide n é chamado de fator primo. É possível determinar a *fatoração prima* de n achando-se o fator primo d e substituindo n pelo quociente n / d , repetindo essa operação até que n seja igual a 1. Utilizando um dos TADs vistos em sala (Lista, Pilha ou Fila) para auxiliá-lo na manipulação de dados, implemente um programa que compute a fatoração prima de um número imprimindo os seus fatores em ordem decrescente. Por exemplo, para $n=3960$, deverá ser impresso $11 * 5 * 3 * 3 * 2 * 2 * 2$. Justifique a escolha do TAD utilizado.
- 7) Utilizando as operações de manipulação de pilhas vistas em sala, uma pilha auxiliar e uma variável do tipo `TipoItem`, escreva um procedimento que remove um item com chave `c` de uma posição qualquer de uma pilha. Note que você não tem acesso à estrutura interna da pilha (topo, item, etc), apenas às operações de manipulação.
- 8) Desenvolva um método para manter duas pilhas dentro de um único vetor linear (um arranjo) de modo que nenhuma das pilhas incorra em estouro até que toda a memória seja usada, e toda uma pilha nunca seja deslocada para outro local dentro do vetor.
- 9) Considero que um estacionamento da Rua Direita, em Ouro Preto, é composto por uma única alameda que guarda até dez carros. Existe apenas uma entrada/saída no estacionamento, e esta extremidade da alameda dá acesso justamente à Rua Direita. Se chegar um cliente para retirar um carro que não seja o mais próximo da saída, todos os carros bloqueando seu caminho sairão do estacionamento. O carro do cliente será manobrado para fora do estacionamento, e os outros carros voltarão a ocupar a mesma sequência inicial.
Escreva um programa que processe um grupo de linhas de entrada. Cada linha de entrada contém um 'E', de entrada, ou um 'S' de saída, e o número da placa do carro. Presume-se que os carros cheguem e partam na mesma ordem que entraram no estacionamento. O programa deve imprimir uma mensagem sempre que um carro chegar ou sair. Quando um carro chegar, a mensagem deve especificar se existe ou não vaga para o carro no estacionamento. Se não houver vaga, o carro partirá sem entrar no estacionamento. Quando um carro sair do estacionamento, a mensagem deverá incluir o número de vezes em que o carro foi manobrado para fora do estacionamento para permitir que os outros carros saíssem.

Exercícios extraídos de (Referências)

- [1] Aaron M. Tenenbaum, Yedidyah Langsam, Moshe J. Augenstein, *Estruturas de Dados Usando C*, Makron Books/Pearson Education, 1995.
- [2] N. Ziviani, F.C. Botelho, Projeto de Algoritmos com implementações em Java e C++, Editora Thomson, 2006.
- [3] F.C. Botelho, Comunicação Pessoal, Belo Horizonte, MG, Brazil, 2003.