



Lista de Exercícios sobre Fila

- 1) Considere uma pilha P vazia e uma fila F não vazia. Utilizando apenas os testes de fila e pilha vazias, as operações Enfileira, Desenfileira, Empilha, Desempilha, e uma variável aux do TipoItem, escreva uma função que inverta a ordem dos elementos da fila.
- 2) 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 uma função 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.
- 3) Considere a implementação de filas usando arranjos “circulares”. Escreva uma função `FuraFila(TipoFila* pFila, TipoItem x)` que insere um item na primeira posição da fila. O detalhe é que seu procedimento deve ser **O(1)**, ou seja, não pode movimentar os outros itens da fila. (observe que neste caso, estaremos desrepeitando o conceito de FILA – primeiro a entrar é o primeiro a sair).
- 4) Existem partes de sistemas operacionais que cuidam da ordem em que os programas devem ser executados. Por exemplo, em um sistema de computação de tempo-compartilhado (“*time-shared*”) existe a necessidade de manter um conjunto de processos em uma fila, esperando para serem executados.
Escreva um programa que seja capaz de ler uma série de solicitações para:
 - a. Incluir novos processos na fila de processo;
 - b. Retirar da fila o processo com o maior tempo de espera;
 - c. Imprimir o conteúdo da lista de processo em determinado momento.Assuma que cada processo é representado por um registro composto por um número identificador do processo.
- 5) Que conjunto de condições é necessário e suficiente para que uma sequência de operações de *Enfileira* e *Desenfileira* sobre uma única fila vazia deixa a fila vazia sem provocar *underflow* (tentativa de executar *Desenfileira* com a fila vazia)? Que conjunto de condições é necessário e suficiente para que essa sequência deixe inalterada uma fila não vazia?



- 6) Se uma fila representada por arranjos (vetores) não é considerada circular, sugere-se que cada operação *Desenfileira* deve deslocar para “frente” todo elemento restante de uma fila. Um método alternativo é adiar o deslocamento até que “tras” seja igual ao último índice do vetor. Quando essa situação ocorre e faz-se uma tentativa de inserir um elemento na fila, a fila inteira é deslocada para “frente”, de modo que o primeiro elemento da fila fique na primeira posição do vetor, ou posição 0, caso a implementação seja em C/C++/Java. Quais são as vantagens desse método sobre um deslocamento em cada operação *Desenfileira*? Quais as desvantagens? Reescreva as funções *Desenfileira*, *Enfileira* e *Vazia* usando esse novo método.
- 7) Como você implementaria uma fila de pilhas? Uma pilha de filas? Uma fila de filas? Escreva rotinas para implementar as operações corretas para cada uma destas estruturas de dados.
- 8) Implemente uma fila de inteiros em C/C++, usando uma implementação por arranjos (um vetor *queue*[100]), onde *queue*[0] e *queue*[1] são usados para representar a posição inicial e final da fila respectivamente e *queue*[2] a *queue*[99] são usados para conter os elementos do vetor. Demonstre como inicializar esse vetor de modo a representar a fila vazia e escreva funções *Desenfileira*, *Enfileira* e *Vazia* para tal implementação.
- 9) Implemente uma fila em C/C++, onde cada item da fila consista em um número variável de inteiros.
- 10) Um **deque** é um conjunto de itens a partir do qual podem ser eliminados e inseridos itens em ambas as extremidades. Chame as duas extremidades de um deque *esq* e *dir*. Como um deque pode ser representado como um vetor em C/C++? Escreva quatro funções em C/C++,
***RemDir*, *RemEsq*, *InsDir*, *InsEsq*,**
para remover e inserir elementos nas extremidades esquerda e direita de um deque. Certifique-se de que as funções funcionem corretamente para o deque vazio e detectem o estouro e o *underflow* (tentativa de remoção quando a fila está vazia). Quais as desvantagens dessa implementação com relação a implementação por encadeamento/alocação dinâmica?
- 11) Suponha que o Beco do Pirão (Praça Tiradentes, Ouro Preto), durante a noite, seja usado como um estacionamento que guarda até 10 carros. Os carros entram pela Praça Tiradentes (PT) e saem pela Rua Barão de Camargos (RBC) (obs: fato fictício gerado a partir de informações extraídas de maps.google.com). Se chegar um cliente para retirar um carro que não esteja estacionado na primeira da RBC, todos os carros entre o carro do cliente e a RBC serão deslocados para fora do estacionamento, o carro do cliente sairá do estacionamento e os outros carros voltarão a entrar pela PT na mesma ordem que saíram pela RBC. Observe que sempre que um carro deixa o estacionamento, todos os carros entre ele e a PT serão deslocados até o começo da RBC de modo que, o tempo inteiro, todos os espaços vazios estão na entrada do estacionamento, ou seja na entrada pela PT.



Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM
Disciplina: Estruturas de Dados I – BCC202
Professor: David Menotti (menottid@gmail.com)
Professora: Emiliana Mara Lopes Simões (simoes.eml@gmail.com)



Escreva um programa que leia um grupo de linhas de entrada. Cada linha contém um 'C', de chegada, e um 'P' de partida, além de um número de placa de licenciamento. Presume-se que os carros chegarão e partirão na ordem especificada pela entrada. O programa deve imprimir uma mensagem cada vez que um carro chegar ou partir. Quando um carro chegar, a mensagem deverá especificar se existe ou não vaga para o carro dentro do estacionamento. Se não existir vaga, o carro esperará pela vaga ou até que uma linha de partida seja lida para o carro. Quando houver espaço disponível, outra mensagem deverá ser impressa. Quando um carro partir, a mensagem deverá incluir o número de vezes que o carro foi deslocado dentro do estacionamento, incluindo a própria partida, mas não a chegada. Esse número será 0 se o carro for embora a partir da linha de espera.

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.