

**ESCOLA POLITÉCNICA DE PERNAMBUCO – POLI**  
**ENGENHARIA DA COMPUTAÇÃO - ECOMP**  
**ESTRUTURAS DE DADOS - 2019.2**

**Lista de exercícios 03 - Filas**  
**30 de agosto de 2019**

**OBS: Para cada questão solucionada, envie em seu projeto os testes efetuados em um método main.**

**(ESCOLHA 05 QUESTÕES PARA RESPONDER)**

**1)** Uma Fila de Prioridades é amplamente utilizada em muitas situações cotidianas; trata-se de uma variante do TAD Fila convencional (FIFO). Nessa nova estrutura, os dados são armazenados conforme a uma chave de prioridade e sua remoção obedece, também, a operação sobre os elementos mais prioritários. Com base nisso, utilizemos uma fila de prioridades para organizar formigas na hora de suas refeições. A estrutura atua sobre o dado Formiga (nome, prioridade) para desenvolver os métodos a seguir:

Grau de prioridade: valor 0 (zero) representa prioridade máxima enquanto seus sucessores decrescem em prioridade. 0 -> 1 -> e assim por diante.

- a) Adicionar formiga à lista de espera de alimentação com base em sua prioridade;
- b) Alimentar (remover da fila) sempre a formiga de prioridade máxima da fila;
- c) Listar todas as formigas pertencentes a seus respectivos níveis de prioridade;
- d) Contar quantas formigas estão na fila para se alimentarem.

**2)** Implemente um TAD Fila Dinâmica para simular uma Loja. A classe Loja possui uma lista (de qualquer tipo) de filas, uma para cada caixa, com inicialmente 3 caixas abertas. As filas possuem "status", podem estar abertas ou fechadas. Implemente os seguintes métodos:

- a) `atendeFila(int n)`, que atende um cliente da fila `n`;
- b) `chegaNovo(Cliente c)`, que insere um novo cliente na fila mais curta. Caso não haja filas, exibir "Filas cheias!";
- c) `abreFila()` e `fechaFila()`, que dá uma lista como aberta ou fechada e exibe "Aberta fila `n`" ou "Fila `n` fechada";

**ESCOLA POLITÉCNICA DE PERNAMBUCO – POLI**  
**ENGENHARIA DA COMPUTAÇÃO - ECOMP**  
**ESTRUTURAS DE DADOS - 2019.2**

c) `veSeAbreFila()`, que vê se alguma fila contém mais de 5 clientes e se há espaço para outra fila. Se houver, abre a fila. Caso não haja espaço, exibir "Sem caixas disponíveis!";

3) Implemente uma fila circular, com todos seus métodos usuais, e use-a em um método que identifica se uma string recebida como parâmetro é um palíndromo.

4) Para um dado número inteiro  $n > 1$ , o menor inteiro  $d > 1$  que divide  $n$  é chamado de fator primo. É possível determinar a fatoraçoão 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. Utilize um TAD Fila para auxiliá-lo na manipulação de dados. Implemente uma função que compute a fatoraçoão prima de um número imprimindo os seus fatores em ordem decrescente.

ENTRADA	SAÍDA
$n = 20$	$5 * 2 * 2$
$n = 3960$	$11 * 5 * 3 * 3 * 2 * 2 * 2$
$n = 6552$	$7 * 3 * 3 * 2 * 2 * 2$

5) Faça uma função que receba três filas, duas já preenchidas em ordem crescente e preencha a última com os valores das duas primeiras em ordem crescente. Utilize o seguinte protótipo: `void preenche (Fila F1, Fila F2, Fila Retorno)`.

6) Deque é uma estrutura de dados na qual os elementos podem ser inseridos ou excluídos de qualquer uma de suas extremidades (do início ou do fim). Assim, crie um deque estático com os seguintes métodos:

(a) - `addEsq(T e)` e `addDir(T e)`

(b) - `remove(T e)`, `popEsq()` e `popDir()`. Os métodos 'pop' além de removerem o último valor da respectiva extremidade, retorna o mesmo.

(c) - `print()`