

# Exercícios: Filas

Ronaldo Fumio Hashimoto

1. Explique a diferença entre Pilha, Fila, e Fila de Prioridades em relação à inserção e remoção de elementos.
2. Em que situações é mais apropriada utilizar uma fila do que uma pilha e vice-versa?
3. Escreva um programa que tenha uma fila cujos registros sejam estruturas que possuem um campo nome e um campo idade, este que representa a prioridade. Quanto maior o valor deste campo, maior a prioridade do registro; ou seja ele fica no começo da fila. Insira  $n$  registros com pessoas de diversas idades na fila e depois imprima-os (nome e idade) em ordem de saída dessas pessoas da fila.
4. Escreva um programa que recebe duas listas encadeadas de inteiros de forma que a inserção e a remoção dos elementos seguem as regras de uma pilha na primeira lista e uma fila na segunda lista. Efetue os seguintes passos:
  - (a) Remova os elementos da primeira lista e insira-os na segunda lista;
  - (b) Depois, remova os elementos da segunda lista e insira-os na primeira lista;
  - (c) Qual deve ser a ordem inicial dos elementos das 2 listas para que no final a segunda lista os elementos fiquem ordenados crescentemente?
5. Faça um programa para simular as filas de um banco (considere 5 filas de atendimento de tamanho 10). O programa deve gerenciar as filas da seguinte forma:
  - (a) Sortear uma das 5 filas e adicionar um cliente novo que chegou ao banco;
  - (b) Sortear uma das filas para ser atendido um cliente. Retirar o cliente da fila sorteada para ser atendido;
  - (c) Mostrar o estado das filas (a quantidade de clientes que tem em cada uma das filas) quando uma delas ou atingir seu valor máximo (10), ou ficar vazia.
6. Considere a seguinte **fila circular** implementada como variável global:

```
# define MAX 200
int ini, fim, Fila[MAX];
```

Implemente as funções abaixo para manipular essa fila:

- (a) `void remove2()` para remover dois elementos da fila, se possível. Caso contrário, remova todos os elementos da fila;
  - (b) `void removeN(int n)` para remover  $n$  elementos da fila, se possível. Caso contrário, remova todos os elementos da fila;
  - (c) `insere2(int a, int b)` para adicionar os dois elementos ( $a$  primeiro e depois  $b$ ) na fila, se possível. Caso contrário, insira até que fila fique cheia;
  - (d) `void pushN (int *v, int n)` para adicionar  $n$  elementos que estão em  $v$  (da posição 0 até a posição  $n - 1$ ) na fila, sendo  $n$  a quantidade de elementos no vetor  $v$ . Caso contrário, insira os elementos até que fila fique cheia;
7. Considere a seguinte **fila circular sem cabeça** implementada como variável global:

```
typedef struct fila {
    int info;
    struct fila * prox;
} Fila;
Fila *ini;
```

Note que como a fila é circular, o último elemento é o que aponta para a primeira célula.

Implemente as funções abaixo para manipular essa fila:

- (a) `insere (int x)` para inserir o elemento `x` na fila;
  - (b) `int remove()` que remove 1 elemento da fila e devolve este elemento via `return`;
  - (c) `int tamanho()` que devolve a quantidade de elementos da fila via `return`.
8. Escreva uma função `compara (Fila *F1, Fila *F2)` para testar se a fila circular apontada por `F1` tem mais elementos do que a fila circular apontada por `F2`. Mas, uma **condição** foi imposta: só é permitido acessar os elementos das filas através de suas operações básicas de `insere` e `remove` da questão anterior.