

Teste 3 – Lista encadeada - Fila

O conceito de *Fila* já foi apresentado como sendo uma estrutura de dados onde o primeiro elemento a entrar na *Fila* é o primeiro a sair. A *Fila* é normalmente representada por uma estrutura que contém um ponteiro para o início da lista encadeada e um ponteiro para o elemento no final lista encadeada. Neste teste, você deve modificar um

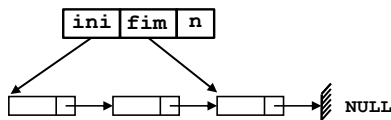


Fig.1

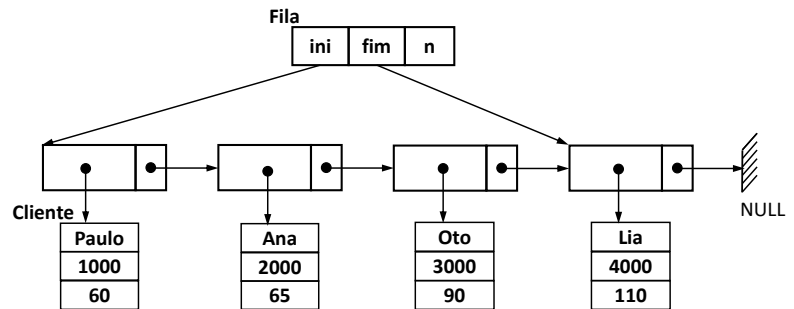


Fig. 2 *Fila* com $n = 4$

pouco esta estrutura para também conter o número total de elementos na *Fila* (i.e. na lista encadeada), conforme o exemplo da Fig. 1.

Considere também que a informação armazenada em cada elemento da *Fila* é um ponteiro para a estrutura **Cliente** que contém o nome do cliente, a quantidade de ações que o cliente quer comprar e o momento que o pedido de compra ocorreu (este momento é especificado em segundos após a abertura da bolsa de valores). Cada vez que um cliente é atendido, o valor de n é atualizado. No exemplo da Fig. 2, Paulo colocou o pedido de compra de 1000 ações, 60 segundos após a abertura da bolsa e Lia, a última a entrar na Fila, fez o pedido de 4000 ações, 110 segundos após a abertura.

Primeiro escreva as seguintes funções de serviço da estrutura *Fila*: **(1)** função para criar *Fila*, que retorna um ponteiro para *Fila*; **(2)** função que testa se uma *Fila* é vazia; **(3)** função *void* com apenas dois parâmetros, um ponteiro para *Fila* e um ponteiro para *Cliente* já alocado estaticamente, que insere o *Cliente* na *Fila*; **(4)** função que recebe uma *Fila*, retira um *Cliente* desta Fila e retorna o seu endereço; **(5)** função que retorna o endereço do primeiro *Cliente* de uma *Fila* (sem retirá-lo); **(6)** função que retorna o tamanho atual de uma *Fila* (i.e. o valor atual de n); **(7)** função que libera completamente uma *Fila* (i.e. libera todas as memórias que foram criadas através de *malloc*). Gerencie adequadamente problemas de memória.

Depois, usando as funções de serviço, escreva uma função que, dada uma *Fila* e uma ocasião de fechamento de compras (em segundos a partir da abertura da bolsa), processa a *Fila* e retorna o total de compras realizadas até antes do fechamento. Além disto, a função disponibiliza o número de clientes processados (i.e., atendidos) e quantos por cento isto representa em relação ao tamanho inicial da *Fila* (estes dois valores devem ser disponibilizados através dos argumentos desta função). No exemplo da Fig. 2, para o fechamento 100 segundos após a abertura, têm-se 3 clientes atendidos (representando 75% de atendimento) e um total de compras de 6000 ações. Esta função também deve cuidar para que todos os clientes com pedidos de compras no momento exato do fechamento ou após permaneçam na Fila. Por fim, esta função imprime os clientes sendo atendidos (com nome e quantidade de ações).

No teste da sua função, na *main*, imprima o momento de fechamento, o total de ações e o número de clientes

remanescentes na Fila quando ela é liberada. Crie cada *Cliente* estaticamente. Use obrigatoriamente todas as funções de serviço. Ver formato de saída na Fig. 3. Use obrigatoriamente valores completamente diferentes dos usados nos exemplos (inclusive o tamanho da Fila). Use apenas o **padrão ANSI C** (o uso de outros padrões anula o teste). Pode usar `//` como comentário. Nomeie o seu arquivo no formato: *nome_ultimoSobrenome_matricula_Testes03_G2*.

```
Cliente 1 (Paulo) atendido, quantidade acoes 1000
Cliente 2 (Ana) atendido, quantidade acoes 2000
Cliente 3 (Oto) atendido, quantidade acoes 3000
Fechamento: 100
Clientes atendidos: 3(75.0%)
Total: 6000

Liberando a fila remanescente com 1 cliente(s)
```

Fig. 3

DISCLAIMER:

Por favor coloque as seguintes linhas de comentário no início do seu código (contendo os seus dados), como sendo a sua identificação e a sua declaração:

```
/*
*****
NOME COMPLETO:
MATRICULA PUC-Rio:
DISCIPLINA: INF1007 TURMA (33A, 33B, 33C, 33D):
DECLARACAO DE AUTORIA:
Declaro que este documento foi produzido em sua totalidade por mim,
sem consultas a outros alunos, professores ou qualquer outra pessoa.
*****
*/
```