

- Prazo para submissão: 11 de junho de 2023
- Peso da atividade: 1

## Fila de Prioridades

### Atividade

O objetivo dessa tarefa é **implementar uma fila de prioridades em um restaurante**, utilizando as estruturas de dados estudadas *max-heap* e *min-heap*.

Trata-se de uma fila de prioridades de clientes, em que cada cliente é uma estrutura que possui os campos obrigatórios **sobrenome (string)** e **prioridade (int)**. A prioridade de cada cliente na fila é indicada pelo campo "prioridade" do próprio objeto.

Nessa fila de prioridades, **é necessário manter a informação tanto do primeiro cliente na fila do restaurante (o próximo a receber uma mesa) quanto do último cliente da fila (que está prestes a desistir e voltar pra casa)**. Para isso, você precisa implementar tanto um **min-heap** quanto um **max-heap**. Ambas as estruturas de dados devem apontar para os mesmos objetos (clientes) na memória.

Para gerenciarmos a lista de espera, faremos uso dos seguintes comandos para chamar operações de manipulação:

- **I s p**: Insere uma nova entrada com os campos sobrenome *s* (string) e prioridade *p*
- **P**: imprime na tela o primeiro e o último cliente da lista de espera, no formato (nome, prioridade), conforme exemplo a seguir:

```
1 [Fulano 10 ... Beltrano 50]
```

- **L**: libera uma mesa para o primeiro da fila e imprime na tela uma string com o seguinte formato:

```
1 Mesa liberada para Sr(a) Fulano.
```

- **D**: remove a última pessoa da fila (que cansou de esperar e voltou para casa). Imprime na tela uma string com o seguinte formato:

```
1 Sr(a) Beltrano desistiu de esperar e foi embora.
```

- **M s p**: busca o cliente com sobrenome *s* e muda sua prioridade para *p*.

## Entrada

A entrada é um arquivo texto composto de:

- um número inteiro  $n$ .
- seguido de  $n$  linhas, cada uma com uma das possíveis instruções.

Exemplo de entrada:

---

```
1 8
2 I Mouse 7
3 I Lightyear 20
4 P
5 M Mouse 25
6 L
7 P
8 D
9 P
```

---

## Saída

A saída é um conjunto de strings, conforme formato especificado acima. Para o exemplo de entrada, temos a seguinte saída:

---

```
1 [Lightyear 20 ... Mouse 7]
2 Mesa liberada para Sr(a) Mouse.
3 [Lightyear 20 ... Lightyear 20]
4 Sr(a) Lightyear desistiu de esperar e foi embora.
5 [ ]
```

---

## Dicas e Orientações Importantes

- Você pode criar uma estrutura de dados **FilaPrioridade** que possui 2 campos principais: um max-heap e um min-heap.
- Você vai ter que saber a posição dos clientes tanto no max-heap quanto no min-heap. Para isso, você pode ter dois campos extras na estrutura de Clientes: uma que guarda a sua posição no min-heap e outra que guarda sua posição no max-heap.
- O tamanho máximo da fila de prioridades é de 100 pessoas.
- Seu programa deve lidar com casos inválidos, mais especificamente:
  - Não pode inserir mais ninguém quando a lista está cheia
  - Não pode remover ninguém de uma lista vazia
  - Não pode modificar a prioridade de alguém que não se encontra na fila.

Em nenhum desses casos, seu programa deve mostrar mensagem de erro, nem parar a sua execução. Você deve, simplesmente, ignorar entradas inválidas.

- Tome bastante cuidado com a formatação das strings de saída. Utilize para isso os exemplos de input que serão dados.
- Nessa atividade, 3 arquivos devem ser entregues: *FilaPrioridade.h*, *FilaPrioridade.c* e *main.c*. Os dois primeiros se referem à implementação da estrutura de dados da fila. O terceiro utiliza essa implementação para a resolução do problema da atividade.
- **Seu programa deve, obrigatoriamente, receber um único parâmetro: um arquivo .txt de entrada que contém a entrada. Nesse caso, você deve utilizar o *argv* para receber do usuário o nome do arquivo .txt. Você também deve utilizar funções de leitura de arquivo para ler a entrada.** Você pode se basear nos códigos disponibilizados na página da disciplina para construir isso.
- Seu programa vai ser compilado da seguinte maneira:

---

```
1 gcc -std=c99 -pedantic -Wall -c -o FilaPrioridade.o FilaPrioridade.c
2 gcc -std=c99 -pedantic -Wall -c -o main.o main.c
3 gcc -o bin main.o FilaPrioridade.o
```

---