

---

## Laboratório #5 - lab5.c

Filas

---

**Prof. Valério Rosset**

### 1 DEFINIÇÃO DO PROBLEMA

Para organizar seus estudos, você decidiu desenvolver uma fila de tarefas. Algumas tarefas são periódicas, ou seja, você deve realizar toda semana. Assim, você decidiu incluir uma opção para mover uma tarefa concluída de volta para o final da fila.

#### TAREFA A

Desenvolva um algoritmo em linguagem C - **utilizando uma fila estática circular** - para implementar uma fila com as tarefas. Ao final da execução, seu algoritmo deverá imprimir a fila de tarefas.

Seu algoritmo deve ser capaz de realizar as seguintes operações:

- **Inserção:** Seu algoritmo deve permitir a inserção de tarefas. Note que a única informação inserida na fila é o ID da tarefa. Como a estrutura utilizada é fila, a inserção ocorre sempre no **final** da fila.
- **Remoção:** Seu algoritmo deve permitir a remoção de tarefas. Como a estrutura utilizada é fila, a remoção ocorre sempre no **início** da fila.
- **Conclusão:** Seu algoritmo deve permitir a conclusão de tarefas. Nesse caso, o elemento do **início** da fila deverá ser deslocado para o **final** da fila.

## ENTRADA

A entrada contém um único conjunto de teste, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). O conjunto de teste é composto por  $L + 1$ ,  $1 \leq L \leq 100$  linhas.

1. A primeira linha contém o número de operações que será realizado  $L$ .
2. Em cada uma das  $L$  linhas seguintes são dadas as informações da operação. São permitidas 3 operações:
  - a) Inserção de Tarefas (1): Nesse caso, a linha é composta por 2 inteiros separados por um espaço. O primeiro valor  $OP$  identifica a operação, no caso 1. O segundo valor  $V$ ,  $1 \leq V \leq 100$ , indica o ID da tarefa. Ou seja, o valor que será inserido na fila.
  - b) Remoção de Tarefas (2): Nesse caso, a linha é composta por 1 inteiro  $OP$  que identifica a operação, no caso 2.
  - c) Conclusão de Tarefas (3): Nesse caso, a linha é composta por 1 inteiro  $OP$  que identifica a operação, no caso 3.

Para as operações de entrada, considere a sintaxe:

```
scanf("%d",&variavel);
```

## SAÍDA

Como saída padrão, seu programa deverá imprimir a fila ao final da execução. Além disso, caso a fila esteja vazia e ocorra alguma tentativa de remoção ou conclusão de tarefas, seu programa deverá imprimir a mensagem: "vazia".

Para as operações de saída, considere a sintaxe:

```
printf("instrucao de saida\n");
```

```
printf("%d ", itemdafila);
```

```
printf("%d\n", ultimoitemdafila);
```

## EXEMPLO 1

**OBSERVAÇÃO:** Nesse exemplo, para demonstrar o funcionamento do algoritmo, a fila é exibida após cada operação. Contudo, seu algoritmo deverá imprimir a fila **somente** ao final da execução, como nos exemplos 2 e 3.

Entrada	Saída
7	4
1 4	7 4
1 7	5 7 4
1 5	4 5 7
3	4 5
2	1 4 5
1 1	5 1 4
3	

### EXEMPLOS 2 E 3

Entrada	Saída
7	5 1 4
1 4	
1 7	
1 5	
3	
2	
1 1	
3	

Entrada	Saída
8	vazia
1 4	5 1
2	
3	
1 5	
1 9	
1 1	
3	
2	

### OBSERVAÇÕES

- É fundamental que o programa imprima apenas o valor de saída.
- Os exercícios devem ser postados no site <http://judge.sjc.unifesp.br>.
- Cada aluno deverá postar apenas um exercício dentro da data limite.
- A resolução submetida deve utilizar a estrutura de dados definida no enunciado do exercício.
- Os plágios serão penalizados. Como não será possível saber quem copiou, todos os casos envolvidos receberão punições iguais, portanto, não divulgue seu código.