

### Lista 12 - Filas

1. Considere uma fila de inteiros. Execute a sequência de operações a seguir na fila, mostrando ao final de cada operação `desenfileire()` o valor que será retornado. Responda a questão sem nenhuma implementação.

```
enfileire(33)
enfileire(7)
enfileire(11)
desenfileire()
desenfileire()
enfileire(2)
desenfileire()
```

Do exercício 2 ao 6, você implementará uma fila (não circular) de inteiros em um vetor. Siga o passo a passo.

2. Declare um vetor de inteiros `f` de tamanho `TAMANHO_FILA` (uma constante com valor 30). Além disso, declare duas variáveis inteiras `s` e `t`, que serão índices que utilizaremos para manipular o início e o fim da fila.
3. Em C/C++, variáveis logo após serem declaradas possuem valores desconhecidos. Tornasse, portanto, imprescindível inicializar a fila após a sua declaração, a colocando em um estado válido. Na função `main`, inicialize a fila de modo que ela comece vazia, ou seja, ajuste as variáveis `s` e `t` para representar uma fila vazia (de acordo com o que vimos na aula teórica).
4. Implemente a função `void enfileire(int f[], int s, int *t, int x)` que insere um elemento ao final da fila. A função deve armazenar o elemento `x` na posição `t` do vetor `f` e, em seguida, incrementar `t` em 1. Caso a fila esteja cheia, deve exibir uma mensagem de erro indicando que ocorreu *transbordamento da fila* (em inglês, *queue overflow*). Obs.: O argumento `t` deve ser passado por referência, por isso o parâmetro é um ponteiro.
5. Implemente a função `int desenfileire(int f[], int *s, int t)` que remove e retorna o elemento no início da fila. A função deve obter o elemento da posição `s` do vetor `f` e, em seguida, atualizar o índice `s`. Caso a fila esteja vazia, deve exibir uma mensagem de erro indicando que ocorreu *esvaziamento da fila* (em inglês, *queue underflow*). Obs.: O argumento `s` deve ser passado por referência, por isso o parâmetro é um ponteiro.
6. Na `main`, teste as operações de enfileirar e desenfileirar para verificar se a sua fila implementada segue corretamente a política FIFO (*First-In-First-Out*). Utilize a sequência de operações do exercício 1 e verifique se o resultado obtido é o esperado.
7. Visualização é sempre fundamental para compreensão de algoritmos. Copie o código desenvolvido e o cole em: <https://pythontutor.com/cpp.html#mode=edit>. Visualize o passo a passo do código construído.
8. Siga a mesma ideia e agora implemente uma fila circular de inteiros em um vetor.

9. Implemente uma fila em uma lista encadeada com célula-cabeça.
10. Implemente uma fila em uma lista duplamente encadeada *circular* com célula-cabeça. O primeiro elemento da fila ficará na segunda célula e o último elemento ficará na célula anterior à cabeça. Para manipular a fila basta conhecer o endereço `ff` da célula-cabeça.