

### Parte I – Filas

1. Criar duas filas e inserir elementos em ambas, aleatoriamente. Numa 3ª fila, fazer o merge das 2 primeiras, esvaziando as filas iniciais. Note que uma pode ser mais longa que outra.
2. Construir uma aplicação que insere  $n$  elementos em uma pilha  $p$ . Depois, usa uma fila  $f$  inicialmente vazia para encontrar um elemento  $x$  em  $p$ . O procedimento deve devolver ou a posição do elemento na pilha ou  $-1$ , caso ele não exista. Além disso, os elementos devem ser devolvidos na ordem original para a pilha. Você pode usar variáveis auxiliares, de qualquer tipo, mas não deve acessar os vetores das estruturas.
3. Um determinado call center atende suas chamadas de acordo com a ordem de chegada e também de acordo com a prioridade da chamada. Construir um controle de call center que insere elementos numa fila de chamada e, a partir dessa fila, distribui as chamadas em filas específicas de acordo com a prioridade:

As prioridades são 1, 2 e 3, logo existem as filas  $f_1$ ,  $f_2$  e  $f_3$ , além da fila de chegada.

Você deve receber um conjunto de chamadas (id e prioridade), inserindo-as na fila de chegada e depois distribuí-las nas suas respectivas filas.

### Parte II – Listas Ligadas (pode ser para listas simples e duplamente ligadas)

1. Inverter os elementos de uma lista
2. Mesclar os elementos de 2 listas (não estão ordenadas)
3. Fazer o split de uma lista em duas
4. Escrever uma função para esvaziar uma lista ligada
  - a. iterativa (um a um)
  - b. eficiente
5. Considere agora que você tem 2 listas com elementos que estão ordenados em cada uma delas. Criar uma 3ª lista com os elementos das duas anteriores, mantendo a ordem.
6. Criar um `insereVirus`: a inserção do elemento  $n$  é sempre no início, depois se o elemento for par, inserir os valores de 1 a  $n$  no final da lista, se o elemento for ímpar, inserir de 1 a  $n$  no início.
7. Criar uma função para ordenar uma lista ligada.
8. Busca um elemento numa lista ligada e devolve a sua posição, isto é, 1, 2, ....
9. Busca um elemento e devolve quantas vezes ele aparece.
10. Busca um elemento na lista e devolve uma lista com as posições do elemento.