



Exercícios

01) Um bar resolveu liberar um número específico de rodadas grátis para seus n clientes presentes no estabelecimento. Esse bar possui X garçons. Cada garçom consegue atender a um número limitado (C) de clientes por vez. Como essas rodadas são liberadas, cada garçom somente vai para a copa para buscar o pedido quando todos os C clientes que ele pode atender tiverem feito o pedido ou não houver mais clientes a serem atendidos. Após ter seu pedido atendido, um cliente pode fazer um novo pedido após consumir sua bebida (o que leva um tempo aleatório) e a definição de uma nova rodada liberada. Uma nova rodada somente pode ocorrer quando foram atendidos todos os clientes que fizeram pedidos. Por definição, nem todos os clientes precisam pedir uma bebida a cada rodada. Implemente uma solução que permita a passagem por parâmetro

- A quantidade de clientes presentes no estabelecimento;
- A quantidade de garçons que estão trabalhando;
- A capacidade de atendimento dos garçons; e
- O número de rodadas que serão liberadas no bar.

Cada garçom e cada cliente devem ser representados por threads, estruturalmente definidos como os pseudocódigos que seguem:

```
thread cliente {
    while (!fechouBar) {
        fazPedido();
        esperaPedido();
        recebePedido();
        consomePedido(); //tempo variável
    }
}

thread garçom{
    while(existemClientesNoBar){
        recebeMaximoPedidos();
        registraPedidos();
        entregaPedidos();
        rodada++; //serve como parâmetro para fechar o bar
    }
}
```

A ordem de chegada dos pedidos dos clientes na fila de pedidos de cada garçom deve ser respeitada. Sua solução não deve permitir que clientes furem essa fila. O garçom só pode ir para a copa quando tiver recebido seus C pedidos. O programa deve mostrar a evolução, portanto planeje bem o que será apresentado. Deve ficar claro o que está acontecendo no bar a cada rodada: os pedidos dos clientes, os atendimentos pelos garçons, os deslocamentos para o pedido, a garantia de ordem de atendimento, etc.

02) Um barbeiro corta o cabelo de qualquer cliente. Se não há clientes, o barbeiro tira uma soneca. Há várias threads, uma para cada cliente. Um cliente aguarda pelo barbeiro se há ao menos uma cadeira vazia na barbearia, caso contrário, o cliente sai da barbearia imediatamente. Se há uma cadeira disponível, então o cliente senta. Se o barbeiro está dormindo, então o cliente acorda-o. Existem $<n>$ cadeiras na barbearia. Faça um programa para a classe **BarbeiroDorminhoco** utilizando monitor.

03) Conserte os problemas

- Roleta
- Saldo bancário