

1) Desenvolva uma função chamada `dequeueN()` para remover N nós a partir do início de uma fila. Retorne apenas o nó que possuir a maior chave, e dê free nos demais.

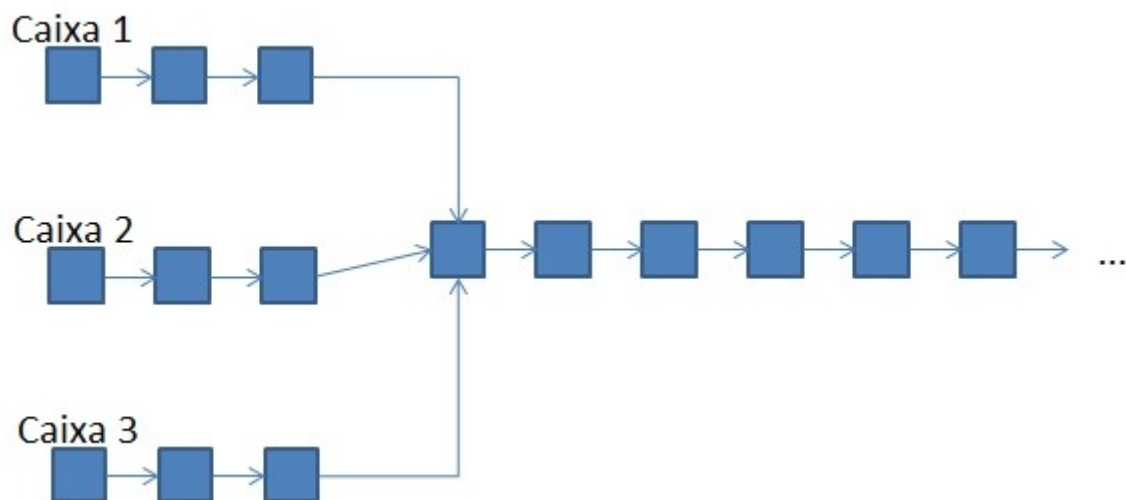
2) Joãozinho e Juquinha gostam muito de jogar bafo. Para quem não sabe, este jogo inicia com uma pilha de figurinhas. A cada jogada, um dos jogadores terá a chance de bater na pilha, tentando virar algumas figurinhas. As que forem viradas passam a pertencer a este jogador.

Joãozinho tem uma técnica apurada, que às vezes é capaz de virar 4 figurinhas com uma só jogada. Entretanto, quando esta técnica falha, nenhuma se vira.

Já a técnica de Juquinha garantidamente vira sempre no mínimo 1 figurinha, mas no máximo 2.

Simule computacionalmente partidas entre Joãozinho e Juquinha, iniciando com 20 figurinhas na pilha e Joãozinho como 1º jogador. Se a técnica dos 2 funcionar ao máximo em todas jogadas, quantas jogadas serão no total? Imprima quantas figuras terá Joãozinho e quantas terá Juquinha. E no caso das técnicas de ambas falharem em todas as jogadas? Imprima também.

3) Um supermercado possui 3 caixas, os quais podem ter filas de no máximo 3 clientes (o que está sendo atendido mais 2). Além desses, todos demais clientes devem formar uma única fila (veja figura abaixo). A medida que um dos caixas libera um cliente, o primeiro da "fila do povão" é encaminhado para uma das filas específicas. Simule o funcionamento destas filas computacionalmente. A liberação dos clientes deve ser feita de forma aleatória (sorteio de 1 a 3, por exemplo).



4) Escreva uma função que inverte a ordem das células de uma lista simplesmente encadeada (a primeira passa a ser a última, a segunda passa a ser a penúltima etc.). Faça isso sem usar espaço auxiliar; apenas altere os ponteiros.

5) Repita o exercício 4 para uma lista duplamente encadeada.

6) Escreva um método para ordenar uma lista. A escolha entre simplesmente ou duplamente encadeada é sua. A escolha de qual método também.

7) Implemente uma busca por chave para uma lista duplamente encadeada sabendo que seus elementos estão ordenados por chave. Se a chave buscada for mais parecida com a chave do 1º nó, faça uma busca sentido início->fim; caso a chave buscada seja mais parecida com a chave do último nó, faça uma busca no sentido fim->início.

8) Copie as linhas abaixo para um arquivo “entrada.txt”:

```
1 1 2 aaa
1 1 5 bbb
1 2 3 ccc
1 4 9 ddd
1 4 8 eee
3 99 99 x
2 3 88 x
3 99 99 x
```

A seguir, crie um programa que leia esse arquivo utilizando no prompt “programa.exe < entrada.txt”. Considerando que cada linha tem 3 números e 1 string no formato “A B C S”, seu programa deve:

- Sempre que A = 1, insira um elemento em uma lista dinâmica duplamente encadeada (LDDE) com chave=C e info=S na posição B;
- Sempre que A = 2, remova o elemento da LDDE com chave = B (ignore C e S, neste caso);
- Sempre que A = 3, imprima todos nós da LDDE (ignore B, C e S, neste caso).

Acrescente outras linhas em “entrada.txt” para testes, caso julgue necessário.