

**Questão 1.** Complete as implementações (com vetor e lista ligada) de pilhas e filas com as operações auxiliares: `inicializa`, `eh_vazia`, `eh_cheia` (para vetor).

**Questão 2.** (Tanenbaum) Escreva um algoritmo para determinar se uma string de caracteres de entrada é da forma:

$$xCy$$

onde  $x$  é uma string consistindo de letras 'A' e 'B' e  $y$  é o inverso de  $x$  (isto é, se  $x$  é a string "ABABBA", então  $y$  deve equivaler a "ABBABA"). Em cada ponto, você só poderá ler o próximo caractere da string.

**Questão 3.** Implemente um tipo abstrato de dados “pilha de inteiros com máximo”, eficientemente, com as seguintes operações: empilhar inteiro, desempilhar inteiro, obter valor máximo. *Cada operação deve ter tempo de execução constante!*

**Questão 4.** Implemente um programa que leia uma expressão aritmética infixada possivelmente com parênteses que reconheça os operadores binários  $+$ ,  $-$ ,  $*$ ,  $^$  e  $/$  e em seguida imprima

- a versão pós-fixa
- a versão pré-fixa
- o resultado da expressão

**Questão 5.** Um *deque* (*double-ended queue*) é um conjunto dinâmico com operações: `insere_inicio`, `insere_fim`, `remove_inicio`, `remove_fim`. Implemente um *deque* utilizando listas ligadas.

**Questão 6. (longo)** Em um banco, há apenas um caixa e todos devem ser atendidos por ordem de chegada. No entanto, pessoas idosas têm prioridade e passam para o início da fila. Considerando que em vários dias o número de idosos é muito grande, foi estipulada a seguinte regra:

1. uma pessoa é atendida na ordem de chegada
2. no máximo 2 idosos podem passar na frente de uma pessoa que não é idosa

Escreva um programa que leia uma sequência de linhas, onde cada linha contém a informação da ordem de chegada e categoria do cliente e imprima a ordem de atendimento (considere que todos chegaram antes de começar o atendimento).

Ex. de entrada:

```
1 geral
2 geral
3 idoso
4 idoso
5 idoso
6 geral
7 idoso
```

Ex. de saída:

```
3 4 1 2 5 7 6
```