

**Exercício 1:**

Fazer um teste de mesa do algoritmo abaixo, deixando explícito os valores de saída em console e a estrutura da pilha

```
Pilha p = new Pilha();

int[] vetor = {100, 200, 1, 50, 39, 44, 25, 16, 99, 45, 33, 18, 102, 92};

int tamanhoVetor = vetor.length;

Para (i = 0 ; i < tamanhoVetor ; i++) {
    Se (pilhaVazia() == verdadeiro) {
        p.push(vetor[i] - 10);
    } Senao Se (vetor[i] mod 5 == 0) {
        p.push(vetor[i] / 5);
    } Senao Se (vetor[i] mod 3 == 0) {
        p.push(vetor[i] * 3);
    } Senao {
        int v1 = p.pop();
        escreva (v1 / 2);
    }
}
```

**Exercício 2:**

Criar um teste de mesa que demonstre a utilização de uma pilha de inteiros para realizar a operação de cálculo de fatorial de um dado número.

Para o teste, deve se usar um número de 0 a 10.

Utilizando as operações possíveis em uma pilha, apresentar o teste de mesa que receba um número de entrada e, utilizando uma pilha vazia e operações, chegar no valor do fatorial do número de entrada.

### Exercício 3:

Baseado na lógica do Exercício 2, criar uma aplicação Java, baseada na biblioteca de pilha de int, que faça:

- 1) Criar uma classe view (Principal.java) que, na main:
  - a. Peça ao usuário um valor inteiro de 0 a 10 (Caso o usuário digite um valor fora desses limites, pedir novamente, até que o valor atenda à solicitação);
- 2) Criar uma classe controller chamada FatController, que tenha um método fatorial(int valor): int e faça:
  - a. Inicialize uma pilha
  - b. Usando a pilha como suporte, calcule o fatorial do valor de entrada
  - c. O método deve retornar esse valor
- 3) O método main da classe Principal.java deve chamar o método fatorial(int valor): long e exibir em console o valor do fatorial.

O método fatorial(int valor): long deve estar baseado nas operações da pilha de inteiros (push(), pop(), size(), top(), isEmpty())