

Lista de Exercícios 1 - Pilhas

Considerando as funções apresentadas em aula, tanto para implementação de pilhas usando **vetores** e **listas encadeadas**, resolva os exercícios abaixo:

1. Desenvolva um algoritmo para testar se uma pilha P1 tem mais elementos que uma pilha P2. Considere que P1 e P2 já existem. O protótipo da função deve ser:

```
int testaMaisElementos(Pilha *P1, Pilha *P2);  
//A função retornará 1 para verdadeiro (P1 > P2) e 0 para falso.
```

2. Desenvolva uma operação para transferir elementos de uma pilha P1 para uma pilha P2 (cópia). Siga o protótipo abaixo:

```
void transferirElementos(Pilha *P1, Pilha *P2, int *flagErro);  
//A função retornará 0 em *flagErro para sucesso e 1 para erro
```

3. Desenvolva um algoritmo para inverter a posição dos elementos de uma pilha P. Você pode criar pilhas auxiliares, se necessário. Mas o resultado precisa ser dado na pilha P.

```
void inverter (Pilha *P);
```

4. Desenvolva um algoritmo para testar se duas pilhas P1 e P2 são iguais. Duas pilhas são iguais se possuem os mesmos elementos, na mesma ordem.

```
int iguais(Pilha *p1, Pilha *p2);  
//retorna 1 para p1 == p2 e 0 para p1 != p2
```

5. Escreva um programa para verificar se uma expressão matemática tem os parênteses agrupados de forma correta, isto é: (1) se o número de parênteses à esquerda e à direita são iguais e; (2) se todo parêntese aberto é seguido posteriormente por um fechamento de parêntese.

Ex1: As expressões ((A+B) ou (A+B violam a condição 1

Ex2: As expressões)A+B(-C ou (A+B)) - (C + D violam a condição 2

6. Escreva um algoritmo para determinar se uma string de caracteres de entrada é da forma xCy, onde x é uma string consistindo das letras A e B e y é o inverso de x, isto é, se x = "ABB", a string deverá conter os caracteres "ABBCBBA".