

Lista de exercícios 3

1. Empregando pilha, implemente um programa para inverter a ordem das letras nas palavras em uma frase. Por exemplo, na frase “teste de pilha”, o programa deverá exibir “etset ed ahlip”.
2. Empregando pilha, construa um programa para ordenar uma sequência de valores inteiros fornecidos pelo usuário. Faça uso de duas pilhas: ORD e AUX. O objetivo é organizar a pilha ORD de modo que nenhum item seja empilhado sobre outro menor, fazendo uso da pilha auxiliar AUX.
 - a) O usuário pode informar quantos valores inteiros ele desejar.
 - b) Ao final, o programa deve mostrar a pilha ORD.
3. Empregando pilha, crie uma função para verificar se uma expressão composta apenas por chaves, colchetes e parênteses está ou não *balanceada*.
 - a) Exemplo de expressões balanceadas: “[{ () () } { }]”, “{ [([{ }])] }”
 - b) Exemplo de expressões não balanceadas: “{ [(}])”, “{ [) () (] }”
4. Desenvolva as seguintes funções:
 - a) Função para testar se duas pilhas P1 e P2 são iguais.
 - b) Função para retornar o número de elementos da pilha que possuem valor par.
5. Uma transportadora precisa de um sistema para gerenciar o carregamento de caixas nos caminhões da empresa. Implemente um programa que atenda a necessidade da empresa, empregando uma pilha definir a ordem do carregamento e descarregamento de caixas de um caminhão.
 - O programa deve possuir um menu com as opções:
 - **Carregar caixa** - empilhar
 - **Descarregar caixa** - desempilhar
 - **Mostrar carga** – mostrar pilha
 - **Mostrar peso total da carga** – criar uma função em Pilha.hpp para realizar a soma do peso de todas as caixas armazenadas na pilha.
 - **Mostrar a quantidade de caixas** – criar uma função em Pilha.hpp para contar a quantidade de caixas na pilha.
 - **Sair do programa**
 - Cada caixa possui um código e peso;

```
struct Caixa
{
    int codigo;
    float peso;
};
```
 - Na estrutura Nó, a variável “dado” deve ser alterada para o tipo Caixa;
 - Alterar as demais funções de Pilha.hpp para o tipo Caixa.
6. Crie uma função para copiar os dados pilha P1 para P2. Esta função deve preservar os dados armazenados em P1, ou seja, após a execução da função, P1 e P2 devem possuir os mesmos elementos e na mesma ordem.
7. Crie um programa para converter número decimais em hexadecimais empregando uma pilha.
8. Crie um programa que implementa uma pilha em que cada nó é uma outra pilha, ou seja, uma pilha de pilhas.