

Atividade de Estruturas de Dados I

Tipo abstrato de dados PILHA

1ª Questão: Implemente um programa que recebe como entrada uma string e indica se a mesma é bem formada ou não. Uma string é bem formada quando seus delimitadores (parênteses, colchetes e chaves) à esquerda possuem os correspondentes à direita na ordem de ocorrência inversa. Exemplos:

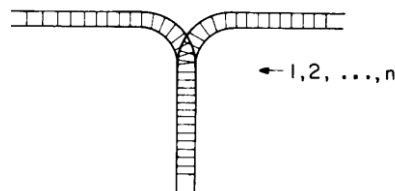
- “abas[ssas{aslk(as)saa}assssssss]” => mal formado
- “mnmk{}asa[ssd]aa{} (as{ss})” => bem formado
- “sas{asd[sdds(sds)sd]sds}das” => mal formado
- “as{assd{fdfff}sdddd}dffsfd{” => mal formado
- “sadsadsds” => bem formado

OBS.: A solução do problema proposto é bem simples se utilizarmos, em nossa lógica, uma pilha. Implemente uma solução para o problema utilizando uma pilha de caracteres.

2ª Questão: Escreva um programa para ler uma frase e determinar se ela é um palíndromo ou não, ou seja, se a frase é soletrada identicamente do início para o fim e do fim para o início. Uma frase válida, para o programa, só deverá conter letras e espaços. Para decidir se uma frase é palíndromo, o programa deve ignorar os espaços. Exemplos de palíndromos: “asa”, “ovo”, “Roma me tem amor, A mala nada na lama”. **OBS.:** Implemente uma solução para o problema utilizando uma pilha de caracteres.

3ª Questão: Escreva um programa para converter um número de notação decimal em um número expresso em um sistema de numeração cuja base é um número entre 2 e 9. A conversão é realizada pela divisão repetida, pela base, à qual o número está sendo convertido, e então se tomando os restos da divisão na ordem inversa. Por exemplo, a conversão para binário do número 6 exige três de tais divisões: $6/2 = 3$ com resto 0, $3/2 = 1$, com resto 1 e, finalmente, $1/2 = 0$ com resto 1. Os restos 0, 1 e 1 são colocados na ordem inversa, de modo que o binário equivalente a 6 é igual a 110. **OBS.:** Utilize uma pilha de inteiros para armazenar os restos das divisões.

4ª Questão: Considere uma rede de desvio ferroviário como abaixo:



Os carros ferroviários estão no trilho situado à direita do desvio e são numerados de acordo com sua posição, da esquerda para a direita. A qualquer momento, um carro pode ser colocado ou retirado de qualquer ponto da composição. Para acrescentar um carro à composição na posição 3, por exemplo, é necessário colocar os carros 1 e 2 no desvio, acrescentar o novo carro (vindo do trilho situado à esquerda do desvio) na posição 3 e recolocar os carros que estavam no desvio, primeiro o 2 e depois o 1. Para retirar o carro situado na posição 4 da composição, é necessário colocar os carros 1, 2 e 3 no desvio, retirar o carro 4 (levando-o para o trilho à esquerda do desvio), e depois recolocar os carros que estavam no desvio, primeiro o 3, depois o 2 e, por último, o carro 1. Implemente um programa que permita controlar a entrada e a saída de carros de uma composição. O programa deverá ter os seguintes procedimentos: (a) Um procedimento para **inserir um carro** na composição em uma dada posição; e (b) Um procedimento para **remover um carro** de uma dada posição da composição. **OBS.:** Implemente a **composição** como sendo uma **pilha de carros**. Um carro tem como atributos o código de identificação do carro e a descrição de sua carga.