

**Universidade Federal do Maranhão**  
**Departamento de Informática**  
**Estrutura de Dados I**  
**Prof. Anselmo Paiva**

**Lista de Exercícios 1 – Pilhas**

1. Use as operações *push*, *pop*, *top* e *empty* para construir operações que façam o seguinte:
  - a. Definir *i* com o segundo elemento a partir do topo da pilha, deixando a pilha sem seus dois elementos superiores.
  - b. Definir *i* com o segundo elemento a partir do topo da pilha, deixando a pilha inalterada.
  - c. Dado um inteiro *n*, definir *i* como o *n*-ésimo elemento a partir do topo da pilha, deixando a pilha sem seus *n* elementos superiores.
  - d. Dado um inteiro *n*, definir *i* como o *n*-ésimo elemento a partir do topo da pilha, deixando a pilha inalterada.
  - e. Definir *i* como o último elemento da pilha, deixando a pilha vazia.
  - f. Definir *i* como o último elemento da pilha, deixando a pilha inalterada. (Dica: use outra pilha auxiliar.)
  - g. Definir *i* como o terceiro elemento a partir do final da pilha.
2. Escreva um algoritmo para determinar se uma string de caracteres de entrada é da forma: *xCy*, onde *x* é uma string consistindo nas letras 'A' e 'B', e *y* é o inverso de *x* (isto é, se *x* = "ABABBA", *y* deve equivaler a "ABBABA"). Em cada ponto, você só poderá ler o próximo caractere da string.
3. Escreva um algoritmo para determinar se uma string de caracteres de entrada é da forma: *a D b D c D ... D z* onde cada string, *a*, *b*, ..., *z*, é da forma da string definida no Exercício 3 (Por conseguinte, uma string estará no formato correto se consistir em qualquer número de strings desse tipo, separadas pelo caractere 'D'.) Em cada ponto, você só poderá ler o próximo caractere da string.
4. Implemente uma fila usando duas pilhas.
5. Faça um procedimento recursivo para procurar por um valor *x* em uma pilha de inteiros, ambos passados como parâmetros, sendo que, ao final, a pilha deverá permanecer intacta.
6. Faça um procedimento *RemoveElemento*( *int* fila *Q*, *int* *x*) que elimina um certo *x* de uma fila *Q* sem alterar a ordem dos demais elementos.
7. Elabore um método para manter duas pilhas dentro de um único vetor linear *[spacesize]* de modo que nenhuma das pilhas incorra em estouro até que toda a memória seja usada, e uma pilha inteira nunca seja deslocada para outro local dentro do vetor. Escreva rotinas em *C*, *push1*, *push2*, *pop1* e *pop2*, para manipular as duas pilhas. (Dica: as duas pilhas crescem na direção da outra.)
8. Utilizando somente operações de empilhar e desempilhar, escreva um programa que remove um item com chave *c* fornecida pelo usuário da pilha. Ao final da execução da função, a pilha deve ser igual à original, exceto pela ausência do item removido.
9. Escreva um algoritmo, usando uma Pilha, que inverte as letras de cada palavra de um texto terminado por ponto (.) preservando a ordem das palavras.

Por exemplo, dado o texto:  
ESTE EXERCICIO E MUITO FACIL.  
A saída deve ser:  
ETSE OICICREXE E OTIUM LICAF

10. Escreva um programa que utilize uma pilha para verificar se uma sequência de caracteres formada somente pelos caracteres '(' e ')' (expressões aritméticas) está com a parentização correta. O programa deve verificar para ver se cada “abre parenteses” tem um “fecha parenteses” correspondente.

Ex.:

Correto: ( ( ) ) ( ( ) ) ( ) ( )

Incorreto: ) ( ( ) ) ( ) ( (

11. 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}asssssssss]” => bem 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

12. Faça um programa que lê três números N, M, P, inteiros. Cria três matrizes A (N, M), B (M, P), C (P, P) e depois permite imprimir C como:

$C = A + B$

$C = A * B$

$C = N * A$

$C = A^t$

13. Faça um programa que lê N, M e os elementos da matriz  $A_{(m,n)}$  e armazena no vetor va e imprimir:

- Os elementos da diagonal principal
- Os elementos da linha L
- Os elementos da coluna C