

1 Considere que os algarismos da base 3, ou seja $b_3=\{0,1,2\}$, formam uma string, que também pode possuir o caractere ' ' (espaço). Agora considere substrings com o seguinte formato: $0^n1^m2^n$, com $m>0$ e $n>0$ (ex: 0001222 00011222 00111122 0000011122222) sempre separadas por um ou mais caracteres espaços. Escreva um programa que leia uma string S e conte quantas substrings possuem o formato padrão indicado.

2 Escreva uma função que recebe uma string S e imprime a string dada na forma inversa, preservando a ordem das palavras. Por exemplo, para a string ESTE EXERCICIO E MUITO FACIL o resultado deve ser ETSE OICIREXE E OTIUM LICAF. (Projeto de Algoritmos em C, Prof. Paulo Feofiloff, USP).

3 Nas expressões aritméticas que estamos acostumados a resolver, os operadores são escritos *entre* os operandos e, neste caso, dizemos que a expressão está escrita na notação *infixa*. Na notação *posfixa*, ou *polonesa*, os operadores são escritos *depois* dos operandos. Eis alguns exemplos de expressões infixas e as correspondentes expressões posfixas:

$(A+B*C)$	ABC*+
$(A*(B+C)/D-E)$	ABC+*D/E-
$(A+B+C*D-E*F*G)$	AB+CD*+EF*G*-
$(A+B*C/D*E-F)$	ABC*D/E*+F-

Escreva uma função que receba duas strings, S1 e S2, contendo, respectivamente, uma expressão escrita na notação infix e posfixa, e indique se elas são matematicamente equivalentes, retornado 0 (falso) ou 1 (verdade), conforme o caso.

4 Valor de expressão polonesa. Suponha que posf é uma string não vazia que guarda uma expressão aritmética em notação posfixa. Suponha que posf contém somente os operadores +, -, *, e / (todos exigem dois operandos). Suponha também que a expressão não tem constantes e que todos os nomes de variáveis na expressão consistem em uma única letra maiúscula. Suponha ainda que temos um vetor valor que dá os valores das variáveis (todos inteiros):

valor[0] é o valor da variável A,
 valor[1] é o valor da variável B,

 valor[26] é o valor da variável z.

Escreva uma função que calcule o valor da expressão posf. Cuidado com divisões por zero! (Projeto de Algoritmos em C, Prof. Paulo Feofiloff, USP).

5 Escreva uma versão iterativa para uma função que simule o comportamento da seguinte função recursiva. Use uma pilha.

```
int f (int n) {
    if (n == 0) return 1;
    else return( n * f(n-1)
}
```

6 A multiplicação de dois números inteiros pode ser feita através de somas sucessivas. Por exemplo, 2×3 é igual a somar o número 2 três vezes, ou o número 3 duas vezes: $2 \times 3 = 2 + 2 + 2 = 3 + 3 = 6$. Implemente um algoritmo recursivo que calcule a multiplicação de dois números inteiros dessa forma.

7 Suponha que os caracteres #, @ e & indiquem respectivamente as operações push, pop e top sobre uma pilha P. Sendo dada uma pilha P, inicialmente vazia, e uma string S formada pelos caracteres #, @ e &, escreva uma função que percorre a string e, ao final, indica quantos elementos existem na pilha. Caso alguma operação não possa ser realizada deve ser dada a mensagem correspondente.

8 Considere duas pilhas P e Q, implementadas no modo dinâmico encadeado. Escreva uma função recursiva que recebe como parâmetros estas pilhas e retorna 1 (verdade) ou 0 (falso), caso elas possuam ou não o mesmo tamanho. Não é Possível a utilização de variáveis locais ou globais à função.

9 Sabe-se que as funções de alocação e liberação de memória gastam muito tempo de processamento. Assim proponha uma solução para uma Pilha de modo que nunca se liberem os nós removidos. Quando um novo elemento for inserido deve-se alocar um novo espaço apenas quando não existir algum espaço já alocado livre. Defina a TAD pilha assim concebida e implemente as 5 operações básicas. Comente os prós e contras da solução proposta.

10 Desenvolva uma função para inverter o elemento do topo com o que aparece abaixo deste, de uma pilha.

11 Desenvolva uma função para intercalar os elementos de duas pilhas P1 e P2 do topo para a base e retornar na pilha P1.

12 Faça uma função que receba duas filas F1 e F2, implementadas no modo estático encadeado, com elementos ordenados de forma crescente e retorne em F1 os elementos de F1 e F2 intercalados em ordem crescente.