

Exercícios Extras - Lista 01

1) Escreva uma função que receba uma cadeia de caracteres e imprima uma tabela que de o número de ocorrências de cada caracter na cadeia. Escreva um programa para testar a função; o programa deve receber a cadeia através da linha de comando.

2) Um **palíndromo** é uma palavra que coincide com sua inversa. Por exemplo, "OSSO". Escreva uma função que decida se uma cadeia de caracteres é ou não um palíndromo. Escreva um programa para testar a função; o programa deve receber a cadeia através da linha de comando.

3) Escreva uma função que decida se uma cadeia de caracteres x é subcadeia de uma cadeia y . Escreva um programa que use a função para fazer o seguinte serviço: ao receber um texto y e uma sequência de palavras x_1, x_2, \dots , imprime as palavras da lista que *não* aparecem no texto. A sequência de palavras é fornecida ao programa como uma sequência de caracteres em que palavras consecutivas são separadas por um ou mais espaços em branco.

4) O programa abaixo procura todas as ocorrências de uma cadeia p em uma cadeia a . Diremos que a cadeia a é um texto e p é uma palavra. Em inglês, p pode ser chamada de *word* ou *pattern*.

```
#include <stdio.h>
#define N 10000

int main(int argc) {
    int i, j, t;
    char a[N], p[N];

    scanf("%s", a);
    scanf("%s", p);

    for (i = 0; a[i] != '\0'; i++) {
        for (j = 0; p[j] != '\0'; j++)
            if (a[i+j] != p[j]) break;
        if (p[j] == '\0') printf("%d ", i);
    }
    printf("\n");
    return 0;
}
```

Escreva uma versão do programa acima que manipule ponteiros em lugar de índices.

5) Qual o valor de $X(5)$?

```
int X(int n) {
    if (n == 1 || n == 2) return n;
    else return X(n-1) + n * X(n-2);
}
```

6) Escreva uma função recursiva que devolva a soma dos elementos de um vetor $v[0..n-1]$ de números inteiros.

7) Escreva uma função recursiva que imprima os elementos de um vetor $v[0..n-1]$ ao contrário, ou seja, primeiro $v[n-1]$, depois $v[n-2]$ e assim por diante.

8) Escreva uma função recursiva que devolva o piso de $\log_2 N$ para qualquer inteiro positivo N . Escreva uma outra função recursiva que devolva o teto de $\log_2 N$ para qualquer inteiro positivo N .

9) Escreva uma função recursiva que receba um inteiro positivo n e devolva a soma dos dígitos decimais de n . Por exemplo, ao receber 1729 sua função deve devolver 19.

10) Dados $t[0]$ e $t[1]$, os números $t[2]$, $t[3]$, \dots são definidos pela relação $t[k] = t[k-2] + t[k-1]$. Critique a seguinte função, que promete calcular $t[n]$ para qualquer $n \geq 2$:

```
int SeqAditiva (int n, int t0, int t1) {
    if (n == 0) return t0;
    return SeqAditiva(n - 1, t1, t0 + t1);
}
```

11) Reescreva o seu programa do exercício 2, mas utilizando ponteiros no lugar de índices.

12) Escreva uma função recursiva que preencha uma matriz de n linhas e 2^n colunas com todas as sequências de n bits na ordem usual de tal forma que o número representado por cada linha seja uma unidade maior que o número representado pela linha anterior. Exemplo com $n = 3$:

```
000
001
010
011
100
101
110
111
```

13) Escreva uma função que faça a indentação de um programa em C. Sua função deve ler um arquivo contendo um programa em C e imprimir o programa devidamente indentado.

14) Escreva um programa que receba uma cadeia de caracteres e gere todas as permutações da cadeia. Por exemplo, ao receber **abc**, o seu programa deve gerar:

```
abc
acb
bac
bca
cab
cba
```

Receita básica para escrever um algoritmo recusivo:

1. Se o problema é pequeno, resolva-o diretamente;
2. Se o problema é grande, *reduza-o* a uma versão menor do mesmo problema e aplique a mesma receita para o problema menor.