

ICMC-USP
Lista de Exercícios Capítulo 2
SCC-0201 - ICC-II

2o. Semestre de 2016 - Turma 2 - Prof. João Luís

1. Mostre, através de teste de mesa, o resultado das seguintes funções:

```
1. int f1(int n)
{
    if (n == 0)
        return (1);
    else
        return(n * f1(n-1));
}
```

Considere as entradas:

- (a) $f1(0)$;
- (b) $f1(1)$;
- (c) $f1(5)$;

```
2. int f2(int n)
{
    if (n == 0)
        return (1);
    if (n == 1)
        return (1);
    else
        return(f2(n-1)+ 2 * f2(n-2));
}
```

Considere as entradas:

- (a) $f2(0)$;
- (b) $f2(1)$;
- (c) $f2(5)$;

```
3. int f3(int n)
{
    if (n == 0)
        printf("`Zero `");
    else
    {
        printf("`%d `",n);
        printf("`%d `",n);
        f3(n-1);
    }
}
```

Considere as entradas:

- (a) $f3(0)$;
- (b) $f3(1)$;

ICMC-USP
Lista de Exercícios 2
SCC-0201 (continuação)

(c) $f3(5)$;

2. Desenvolva algoritmos recursivos para os seguintes problemas:

1. Impressão de um número natural em base binária.
2. Multiplicação de dois números naturais, através de somas sucessivas (Ex.: $6 * 4 = 4 + 4 + 4 + 4 + 4 + 4$).
3. Soma de dois números naturais, através de incrementos sucessivos (Ex.: $3 + 2 = ++(++3)$).
4. Multiplicação de dois números naturais, através de incrementos sucessivos.
5. Cálculo de $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}$.
6. Cálculo de $\frac{2}{4} + \frac{5}{5} + \frac{10}{6} + \frac{17}{7} + \frac{26}{8} + \dots + \frac{(n^2+1)}{(n+3)}$.
7. Inversão de uma string.
8. Gerador da sequência dada por:
 - $F(1) = 1$
 - $F(2) = 2$
 - $F(n) = 2 * F(n - 1) + 3 * F(n - 2)$.
9. Gerador da sequência: 1, 2, 5, 12, 29, 68, 165,
10. Gerador da sequência: 0, 1, 1, 2, 3, 7, 16, 65, 321,
11. Gerador de Sequência de Ackerman:
 - $A(m, n) = n + 1$, se $m = 0$
 - $A(m, n) = A(m - 1, 1)$, se $m \neq 0$ e $n = 0$
 - $A(m, n) = A(m - 1, A(m, n - 1))$, se $m \neq 0$ e $n \neq 0$.
12. A partir de um vetor de números inteiros, calcule a soma e o produto dos elementos do vetor.
13. Gerador de máximo divisor comum (mdc):
 - $mdc(x, y) = y$, se $x \geq y$ e $x \bmod y = 0$
 - $mdc(x, y) = mdc(y, x)$, se $x < y$
 - $mdc(x, y) = mdc(y, x \bmod y)$, caso contrário.
14. Verifique se uma palavra é palíndromo (Ex. *aba*, *abcba*, *xyzzyx*).
15. Dado um número n , gere todas as possíveis combinações com as n primeiras letras do alfabeto. Ex.: $n = 3$. Resposta: ABC, ACB, BAC, BCA, CAB, CBA.
16. Gere todas as possíveis combinações para um jogo da MegaSena com 6 dezenas.

3. Verifique o que as funções dos algoritmos abaixo imprimem e retornam:

ICMC-USP
Lista de Exercícios 2
SCC-0201 (continuação)

```
1. func (int n)
{
    if (n == 0)
        printf("`fim'");
    else
    {
        printf(n);
        func(n-1);
    }
}
```

```
2. func (int n)
{
    if (n == 0)
        printf("`fim'");
    else
    {
        func(n-1);
        printf(n);
    }
}
```

```
3. func (int n)
{
    if (n == 0)
        printf("`fim'");
    else
    {
        printf(n);
        func(n-1);
        printf(n);
    }
}
```

```
4. func (int n)
{
    if (n == 0)
        printf("`fim'");
    else
    {
        func(n-1);
        printf(n);
        func(n-1);
    }
}
```

4. Compare e explique o funcionamento dos algoritmos do exercício anterior.
5. Escreva um programa recursivo em C para classificar (ordenar) um vetor a como segue:
 - Seja k o índice do elemento do meio do vetor.
 - Classifique os elementos até, e inclusive, $a[k]$.
 - Classifique os elementos depois de $a[k]$.
 - Combine os dois subvetores num único vetor classificado.

Este método é chamado de **classificação mesclada**.

6. Determine o que a seguinte função recursiva em C calcula. Escreva uma função iterativa para atingir o mesmo objetivo.

```
func (int n)
{
    if (n == 0)
        return(0);
    return(n + func(n-1));
}
```

7. Defina uma **sequência de Fibonacci generalizada**, de f_0 a f_1 como sequência $fibg(f_0, f_1, 0)$, $fibg(f_0, f_1, 1)$, $fibg(f_0, f_1, 2)$, ..., onde:
 - $fibg(f_0, f_1, 0) = f_0$
 - $fibg(f_0, f_1, 1) = f_1$
 - $fibg(f_0, f_1, n) = fibg(f_0, f_1, n - 1) + fibg(f_0, f_1, n - 2)$, se $n > 1$.

Escreva uma função recursiva em C para calcular $fibg(f_0, f_1, n)$. Descubra um método iterativo para calcular essa função.

Referências

- [1] Nakamiti, G., *Listas de Exercícios de Estruturas de Dados II*, Engenharia de Computação. PUC-Campinas, 2007.
- [2] Tenenbaum, A. M., Langsam, Y., Augestein, M. J., *Estruturas de Dados Usando C*. Makron Books, 1995.