

Lista de Exercícios 2: Recursividade

*Professor: Moacir Pereira Ponti Jr.**PAE(s): Pâmela/Paulo Henrique*

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

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

Considere as entradas:

- i. $f1(0)$;
- ii. $f1(1)$;
- iii. $f1(5)$;

```
(ii) 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:

- i. $f2(0)$;
- ii. $f2(1)$;
- iii. $f2(5)$;

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

Considere as entradas:

- i. $f3(0)$;
- ii. $f3(1)$;
- iii. $f3(5)$;

2. Desenvolva algoritmos recursivos para os seguintes problemas:

- (i) Impressão de um número natural em base binária.
- (ii) Multiplicação de dois números naturais, através de somas sucessivas (Ex.: $6 * 4 = 4 + 4 + 4 + 4 + 4 + 4$).
- (iii) Soma de dois números naturais, através de incrementos sucessivos (Ex.: $3 + 2 = + + (+ + 3)$).
- (iv) Multiplicação de dois números naturais, através de incrementos sucessivos.
- (v) Cálculo de $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}$.
- (vi) Cálculo de $\frac{2}{4} + \frac{5}{5} + \frac{10}{6} + \frac{17}{7} + \frac{26}{8} + \dots + \frac{(n^2+1)}{(n+3)}$.
- (vii) Inversão de uma string.
- (viii) Gerador da sequência dada por:
 - $F(1) = 1$
 - $F(2) = 2$
 - $F(n) = 2 * F(n - 1) + 3 * F(n - 2)$.
- (ix) Gerador da sequência: 1, 2, 5, 12, 29, 68, 165,
- (x) Gerador da sequência: 0, 1, 1, 2, 3, 7, 16, 65, 321,
- (xi) 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 $n \neq 0$.
- (xii) A partir de um vetor de números inteiros, calcule a soma e o produto dos elementos do vetor.
- (xiii) 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.
- (xiv) Verifique se uma palavra é palíndromo (Ex. *aba*, *abcba*, *xyzzyx*).
- (xv) 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.
- (xvi) 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:

```

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

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

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

(iv) 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] Parte deste material foi adaptado das listas de exercícios do Prof. João Luís Garcia Rosa, ICMC/USP.