

# Exercícios

## AEDS-II: Recursividade

Prof. Renê Rodrigues Veloso

4 de agosto de 2014

1. Apresente o resultado das seguintes funções:

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

Considere as seguintes entradas:

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

(b) `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 seguintes entradas:

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

(c) `int f3(int n)`  
{  
    `if (n == 0)`  
        `printf("Zero ");`

```

else
{
    printf("%d ",n);
    printf("%d ",n);
    f3(n-1);
}
}

```

Considere as seguintes entradas:

- i. f3(0);
  - ii. f3(1);
  - iii. f3(5);
2. Desenvolva algoritmos recursivos pra os seguintes problemas:
- (a) Impressão de um número inteiro positivo em base binária.
  - (b) Multiplicação de dois números inteiros positivos, através de somas sucessivas. Ex.:  $6 + 4 = 4 + 4 + 4 + 4 + 4 + 4$ .
  - (c) Soma de dois números inteiros positivos, através de incrementos sucessivos. Ex.:  $3 + 2 = ++ (+ + 3)$ .
  - (d) Inversão de uma string.
  - (e) Calcular o 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
  - (f) Verificar se uma palavra é palíndromo. (Ex.: aba, abcba, xyzzyx)
  - (g) 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.
  - (h) Gere todas as possíveis combinações para um jogo da MegaSena com 6 dezenas.

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

```

(a) func (int n)
{
    if (n == 0)
        printf("fim");
    else
    {
        printf("%d ",n);
        func(n-1);
    }
}

```

```

(b) func (int n)
{
    if (n == 0)
        printf("fim");
    else
    {
        func(n-1);
        printf("%d ",n);
    }
}

(c) func (int n)
{
    if (n == 0)
        printf("fim");
    else
    {
        printf("%d ",n);
        func(n-1);
        printf("%d ",n);
    }
}

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

```

4. Defina uma sequência de Fibonacci generalizada, de  $f_0$  a  $f_1$  como a sequência  $fibg(f_0, f_1, 0), fibg(f_0, f_1, 1), fibg(f_0, f_1, 2), \dots$ , 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.