

# Lista de Exercícios de CES-11

## CTA - ITA - IEC

**Importante:** Não vale nota, ou seja, não é preciso entregar!

### A. Passagem de parâmetros, escopo de variáveis, recursão

1. Analise o código abaixo. Qual é a sua saída?

```
#include <stdio.h>
#include <conio.h>

int i = 10, j = 20;

void g (int i, int j, int k, int l) {
    printf ("%7d%7d%7d%7d\n\n", i, j, k, l);
}

void f (int *m, int *n, int p, int q) {
    int k, l;
    g (*m, *n, p, q); k = 5; l = 8;
    *m = 500; *n = 600; p = 800; q = 700;
    g (i, j, k, l); g (*m, *n, p, q);
}

void main () {
    int i, j, k, l;
    i = 6; j = 7; k = 8; l = 9; g (i, j, k, l);
    {
        int j, k;
        i = 44; j = 45; k = 46; l = 47;
        g (i, j, k, l); f (&j, &k, i, l);
    }
    g (i, j, k, l);
    getch();
}
```

2. A sequência de *Fibonacci* é definida por: 1, 1, 2, 3, 5, 8, 13, 21, ... Repare que os dois primeiros termos são o número 1, e os demais são a soma dos dois termos predecessores. Escreva um algoritmo iterativo que calcule e imprima o  $n$ -ésimo termo dessa sequência. Idem, mas com um algoritmo recursivo **Fib(n)**.

3. Simule a pilha de execução para **Fib(4)**. Seu computador consegue calcular **Fib(50)**? Por quê?

4. Escreva uma função recursiva que receba um número inteiro  $n > 0$  e calcule o valor da série  $1 + 1/2! + 1/3! + \dots + 1/n!$ .

5. Simule a pilha de execução do código abaixo. O que ele faz?

```
#include <stdio.h>
#include <conio.h>

int rec (int x[], int n) {
    if (n < 0) return 0;
    if (x[n] > 0) return x[n] + rec (x, n-1);
    else return rec (x, n-1);
}

void main() {
    int v[5] = {-1, 0, 1, -2, 3};
    printf("%d\n", rec(v,4));
    getch();
}
```

6. Escreva uma função recursiva que imprima o elemento de maior valor presente em um vetor de n inteiros. O vetor também deve ser recebido como parâmetro, e seus índices variam de 0 a n-1.

7. Escreva uma função recursiva que recebe dois números inteiros, e retorna o resultado do somatório de todos os números contidos no intervalo aberto delimitado por esses valores.

8. Escreva uma função recursiva que recebe um número inteiro  $n > 0$  e imprime as sequências de n a 1 e de 1 a n.

Exemplo para  $n=10$ : 10 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 10

9. Escreva uma função recursiva que recebe um número decimal positivo e retorna seu valor em binário.

a) Elabore uma versão em que o resultado é escrito na tela.

b) Elabore uma versão em que a função retorna um número inteiro, formado apenas por 0's e 1's, que corresponde ao resultado calculado.

10. Escreva uma função que recebe um número inteiro  $n > 0$  e gere recursivamente uma sequência com todas as representações de n bits, de tal modo que, entre duas sucessivas, haja um único bit distinto.

*Dica (Gray Code):* A solução para n bits pode ser obtida do seguinte modo:

a) Gere a solução para n-1 bits.

b) Adicione um bit 0 mais à esquerda de cada elemento, formando a sequência x.

c) Gere a mesma solução para n-1 bits, mas agora em ordem invertida.

d) Adicione um bit 1 mais à esquerda de cada elemento, formando a sequência y.

e) Concatene a sequência x com a y.