UNIRIO – Universidade Federal do Estado do Rio de Janeiro
Bacharelado em Sistemas de Informação
Técnicas de Programação 1 – 01/2020
Prof. Jobson Massollar

7ª Lista de Exercícios
Recursividade

Para cada exercício, crie um programa em C/C++ e implemente a função **main** para testar a sua função.
Antes de implementar a função recursiva, escreva a definição recursiva do problema.

1. Implemente a função recursiva `hanoi` a partir do algoritmo apresentado em sala de aula. Para simular a operação de mover o disco, imprima a mensagem "mover o disco da haste X para a haste Y".

2. Implemente a função recursiva `soma` que recebe um número inteiro **n** e retorna a soma de **1..n**.

3. Implemente a função recursiva `soma` que recebe dois números inteiros **a** e **b** e retorna **a x b**.

4. Implemente a função recursiva `soma_digitos` que recebe um número inteiro **n** e retorna a soma dos seus dígitos. Por exemplo: se **n = 1234**, então a função retorna **10**.

5. A função de Ackermann é definida para valores inteiros e não negativos **m** e **n** da seguinte forma:

$$A(m,n) = \begin{cases} n+1 & \text{se } m = 0 \\ A(m-1,1) & \text{se } m > 0 \text{ e } n = 0 \\ A(m-1, A(m,n-1)) & \text{se } m > 0 \text{ e } n > 0. \end{cases}$$

   Implemente a função recursiva `ack` que implementa a função de Ackermann. Calcule ack(3, 2).

6. Implemente uma função recursiva `conta_ocorrencias` que retorna quantas vezes um dígito **k** ocorre em um número natural **n**. Por exemplo, o dígito **2** ocorre **3** vezes em **762021192**.

7. Implemente de forma recursiva a função

```
int mod(int n, int m)
```

   que calcula o resto da divisão inteira (mod) de **n** por **m**. A definição de resto da divisão é dada por:

   MOD(x, y) = MOD (x - y, y), se x > y

   MOD (x, y) = x, se x < y

   MOD (x, y) = 0, se x = y

8. Implemente de forma recursiva a função

```
int div(int n, int m)
```

que calcula o quociente da divisão inteira de **n** por **m**. A definição de quociente da divisão é dada por:

DIV(x, y) = 1 + DIV(|x| - |y|, |y|), se |x| > |y|

DIV (x, y) = 0, se |x| < |y|

DIV (x, y) = 1, se |x| = |y|

9. Implemente de forma recursiva a função

```
int mdc(int n, int m)
```

que calcula o MDC dos números inteiros positivos **n** e **m**. Caso **n** ou **m** não seja um inteiro positivo a função deverá retornar **-1**. A definição de MDC é dada por:

MDC(x, y) = MDC(x - y, y), se x > y

MDC(x, y) = MDC(y, x), se x < y

MDC(x, y) = x, se x = y

Desafios

10. Implemente uma função recursiva que recebe um número inteiro **n** e inverte esse número. Por exemplo: se **n = 123**, então a função retorna **321**

11. Implemente uma função recursiva que inverte as posições de um vetor de inteiros, ou seja, o primeiro vai para a última posição, o segundo para penúltima e assim por diante.

12. Um problema típico em ciência da computação consiste em converter um número da sua forma decimal para a forma binária. Implemente a função recursiva `converte_binario` que recebe um número positivo **n** e imprime esse número em notação binária.

13. Crie a função recursiva `converte_decimal` que faz o inverso da função do exercício anterior, ou seja, recebe um número positivo **n** em notação binária e imprime esse número em notação decimal.