

## Aula Prática 8

Data de entrega: **até às 23:59 de 30/01/2022**

---

João Guilherme Maia de Menezes  
Jefersson Alex dos Santos

24/01/2022

### Instruções para Submissão

Na aula prática de hoje, você terá que elaborar 7 programas para resolver problemas diversos, conforme descrito abaixo. Cada uma das soluções deverá ser implementada em seu próprio arquivo com extensão `.py`. Por exemplo, a solução para o problema 1 deverá ser implementada em um arquivo chamado `problema1.py`, a solução para o problema 2 deverá ser implementada no arquivo `problema2.py` e assim por diante. Finalmente, submeta cada um dos arquivos pelo Moodle.

**Dica:** se você tiver problemas com caracteres especiais (caracteres com acentos, por exemplo), adicione a linha abaixo na primeira linha de todos os arquivos `*.py`

```
# -*- coding: utf-8 -*-
```

### Problema 1

Uma soma pode ser definida recursivamente conforme abaixo:

$$\sum_{k=m}^n k = \begin{cases} m & \text{se } n = m \\ m + \sum_{k=m+1}^n k & \text{se } n > m \end{cases}$$

Implemente uma **função recursiva** chamada `soma` que receba dois parâmetros  $m$  e  $n$  e retorne o valor da soma conforme a definição acima. Em seguida, escreva um programa para testar essa função. O programa deve ler do usuário os valores de  $m$  e  $n$ , chamar a função e imprimir na tela o valor retornado pela função.

**Observação:** as mensagens exibidas para o usuário deverão ser exatamente como apresentado abaixo (mensagens exibidas com os comandos `input()` e `print()`).

**Exemplo de execução do programa:**

```
Digite m: 1
Digite n: 4
10
```

## Problema 2

Escreva uma **função recursiva** chamada `power` que receba como parâmetros dois inteiros positivos  $k$  e  $n$  e retorne o resultado de  $k^n$ . Na sua implementação, você **deve** utilizar apenas multiplicações. Você **não deve** utilizar o operador `**`. Em seguida, escreva um programa para testar essa função. O programa deve ler do usuário os valores de  $k$  e  $n$ , chamar a função e imprimir na tela o valor retornado pela função.

**Observação:** as mensagens exibidas para o usuário deverão ser exatamente como apresentado abaixo (mensagens exibidas com os comandos `input()` e `print()`).

**Exemplo de execução do programa:**

```
Digite k: 2
Digite n: 3
8
```

## Problema 3

O máximo divisor comum (MDC) dos inteiros  $x$  e  $y$  é o maior divisor inteiro comum a  $x$  e  $y$ . Por exemplo, o MDC de 16 e 36 é 4, enquanto que o MDC de 30 e 54 é 6. Escreva uma **função recursiva** chamada `mdc` que retorne o máximo divisor comum de  $x$  e  $y$ . Em seguida, escreva um programa para testar essa função. O programa deve ler do usuário os valores de  $x$  e  $y$ , chamar a função e imprimir na tela o valor retornado pela função. O `mdc` de  $x$  e  $y$  é definido como segue: se  $y$  é igual a 0, então `mdc(x,y)` é  $x$ ; caso contrário, `mdc(x,y)` é `mdc(y, x%y)`, onde `%` é o operador resto.

**Observação:** as mensagens exibidas para o usuário deverão ser exatamente como apresentado abaixo (mensagens exibidas com os comandos `input()` e `print()`).

**Exemplo de execução do programa:**

```
Digite x: 16
Digite y: 36
4
```

## Problema 4

Implemente uma **função recursiva** chamada `soma` que receba como parâmetro um número inteiro positivo  $N$  e retorne o somatório dos números de 1 a  $N$ . Em seguida, escreva um programa para testar essa função. O programa deve ler do usuário o valor de  $N$ , chamar a função e imprimir na tela o valor retornado pela função.

**Observação:** as mensagens exibidas para o usuário deverão ser exatamente como apresentado abaixo (mensagens exibidas com os comandos `input()` e `print()`).

**Exemplo de execução do programa:**

```
Digite N: 4
10
```

## Problema 5

Implemente uma **função recursiva** chamada `imprime_naturais` que receba como parâmetro um número inteiro positivo  $N$  e imprima na tela todos os números naturais de 0 até  $N$  em ordem crescente. Em seguida, escreva um programa para testar essa função. O programa deve ler do usuário o valor de  $N$  e chamar a função.

**Observação:** as mensagens exibidas para o usuário deverão ser exatamente como apresentado abaixo (mensagens exibidas com os comandos `input()` e `print()`).

**Exemplo de execução do programa:**

```
Digite N: 4
0
1
2
3
4
```

## Problema 6

Implemente uma **função recursiva** chamada `imprime_naturais_pares` que receba como parâmetro um número inteiro positivo  $N$  (par ou ímpar) e imprima todos os números pares de 0 até  $N$  em ordem crescente. Observe que o  $N$  só será impresso se o mesmo for par. Em seguida, escreva um programa para testar essa função. O programa deve ler do usuário o valor de  $N$  e chamar a função.

**Observação:** as mensagens exibidas para o usuário deverão ser exatamente como apresentado abaixo (mensagens exibidas com os comandos `input()` e `print()`).

**Exemplo de execução do programa:**

Digite N: 4

0

2

4

## Problema 7

Implemente uma **função recursiva** chamada `imprime_naturais_impares` que receba como parâmetro um número inteiro positivo  $N$  (par ou ímpar) e imprima todos os números ímpares de 1 até  $N$  em ordem decrescente. Observe que o  $N$  só será impresso se o mesmo for ímpar. Em seguida, escreva um programa para testar essa função. O programa deve ler do usuário o valor de  $N$  e chamar a função.

**Observação:** as mensagens exibidas para o usuário deverão ser exatamente como apresentado abaixo (mensagens exibidas com os comandos `input()` e `print()`).

**Exemplo de execução do programa:**

Digite N: 4

3

1