

Aula Prática 4

Instruções para Submissão

Na aula prática de hoje, você terá que elaborar 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

Escreva uma função chamada `populacao` que receba como parâmetros o número de habitantes e a taxa anual de crescimento (em porcento) da população de uma cidade A, o número de habitantes e a taxa anual de crescimento da população de uma cidade B, e retorna o número de anos necessários para que a população da cidade A ultrapasse ou iguale a população da cidade B, mantidas as taxas de crescimento.

Observação 1: O nome da função deve ser exatamente como especificado acima e seu arquivo deve obrigatoriamente se chamar `problema1.py`.

Observação 2: Seu programa deve conter apenas a função descrita acima. Você não precisa realizar a entrada e saída de dados (não precisa usar as funções `input()` e `print()`).

Exemplo de execução do programa:

Digite a população da cidade A: **80000**
Digite a taxa de crescimento da população da cidade A: **3**
Digite a população da cidade B: **200000**
Digite a taxa de crescimento da população da cidade B: **1.5**
63

Problema 2

Desenvolva um gerador de tabuada, capaz de gerar a tabuada de qualquer número inteiro entre 1 e 10. O usuário deve informar de qual número ele deseja ver a tabuada.

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 um inteiro: 5
Tabuada de 5:
5 X 1 = 5
5 X 2 = 10
...
5 X 10 = 50
```

Problema 3

O Sr. Manoel Joaquim possui uma grande loja de artigos de R\$ 1,99, com cerca de 10 caixas. Para agilizar o cálculo de quanto cada cliente deve pagar ele desenvolveu uma tabela que contém o número de itens que o cliente comprou e ao lado o valor da conta. Desta forma, a atendente do caixa precisa apenas contar quantos itens o cliente está levando e olhar na tabela de preços. Você foi contratado para desenvolver o programa que monta esta tabela de preços, que conterá os preços de 1 até 50 produtos.

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:

```
Loja Quase Dois - Tabela de preços
1 - R$ 1.99
2 - R$ 3.98
...
50 - R$ 99.50
```

Problema 4

O Sr. Manoel Joaquim acaba de adquirir uma panificadora e pretende implantar a metodologia da tabelinha, que já é um sucesso na sua loja de 1,99. Você foi contratado para desenvolver o programa que monta a tabela de preços de pães, de 1 até 50 pães, a partir do preço do pão informado pelo usuário.

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 o preço do pão: 0.18
1 - R$ 0.18
2 - R$ 0.36
...
50 - R$ 9.00
```

Problema 5

Faça um programa que leia um número inteiro e imprima na tela se ele é ou não um número primo. Um número primo é aquele que é divisível somente por ele mesmo e por 1.

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 um inteiro n: 2
É primo
```

Exemplo de execução do programa:

```
Digite um inteiro n: 4
Não é primo
```

Problema 6

A série de Fibonacci é formada pela sequência 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,... Faça um programa que imprima na tela o n -ésimo termo, onde n é um inteiro informado pelo usuário.

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 um inteiro n: 7
13
```

Exemplo de execução do programa:

```
Digite um inteiro n: 3
2
```

Problema 7

Escreva um programa que faça a leitura de vários números inteiros até que se digite um número negativo. Em seguida, imprima na tela a soma de todos os números digitados (com exceção do número negativo).

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

Exemplo 1 de execução do programa:

Digite um número: 1
Digite um número: 2
Digite um número: 3
Digite um número: 4
Digite um número: -5
Soma: 10

Exemplo 2 de execução do programa:

Digite um número: 1
Digite um número: 1
Digite um número: 1
Digite um número: -1
Soma: 3

Exemplo 3 de execução do programa:

Digite um número: 1
Digite um número: -1
Soma: 1

Problema 8

Escreva uma função chamada `quantidade_pares` que receba como parâmetros dois números inteiros (`inicio` e `fim`) representando o início e o fim de um intervalo. A função deve então retornar a quantidade de números pares no intervalo `[inicio, fim]`. Você pode assumir que os parâmetros passados para a função sempre obedecerão a condição de que $inicio \leq fim$.

Observação 1: O nome da função deve ser exatamente como especificado acima e seu arquivo deve obrigatoriamente se chamar `problema8.py`.

Observação 2: Seu programa deve conter apenas a função descrita acima. Você não precisa realizar a entrada e saída de dados (não precisa usar as funções `input()` e `print()`).

Exemplo 1 de execução do programa:

Digite o início do intervalo: 1
Digite o final do intervalo: 10
Quantidade de números pares: 5

Exemplo 2 de execução do programa:

Digite o início do intervalo: 2

Digite o final do intervalo: 2

Quantidade de números pares: 1

Problema 9

Escreva um programa que faça a leitura de vários números inteiros até que se digite um número negativo. Em seguida, imprima na tela a soma de todos os números pares e de todos os números ímpares digitados (com exceção do número negativo).

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

Exemplo 1 de execução do programa:

Digite um número: 1

Digite um número: 2

Digite um número: 3

Digite um número: 4

Digite um número: 5

Digite um número: 6

Digite um número: -1

Soma pares: 12**Soma ímpares: 9****Exemplo 2 de execução do programa:**

Digite um número: 1

Digite um número: 1

Digite um número: -5

Soma pares: 0**Soma ímpares: 2****Exemplo 3 de execução do programa:**

Digite um número: 2

Digite um número: 2

Digite um número: -2

Soma pares: 4**Soma ímpares: 0**

Problema 10

Escreva uma função chamada `soma_divisores` que recebe como **parâmetro** um número inteiro n e **retorna** a soma de todos os divisores desse número, com exceção dele próprio. Ou seja, considere o intervalo $[1, n)$.

Observação 1: O nome da função deve ser exatamente como especificado acima e seu arquivo deve obrigatoriamente se chamar `problema10.py`.

Observação 2: Seu programa deve conter apenas a função descrita acima. Você não precisa realizar a entrada e saída de dados (não precisa usar as funções `input()` e `print()`).

Exemplo 1 de execução do programa:

Digite um número: **100**
Resultado: **117**

Exemplo 2 de execução do programa:

Digite um número: **39**
Resultado: **17**