

## 1. Validador de Sudoku

Implemente um programa que verifique se uma solução de Sudoku 9×9 é válida. O Sudoku deve seguir as seguintes regras:

- Cada linha deve conter todos os números de 1 a 9, sem repetição.
  - Cada coluna deve conter todos os números de 1 a 9, sem repetição.
- a) Crie uma função chamada ***isValidGroup*** que:
- a) Receba como argumento uma lista de 9 números.
  - b) Verifique se contém exatamente os números de 1 a 9, sem repetição.
  - c) Retorne True se for válido, ou False caso contrário.
- b) Crie uma função chamada ***isValidSudoku*** que:
- Verifique todas as linhas do tabuleiro.
  - Verifique todas as colunas.
  - Retorne True se todas as condições forem satisfeitas, ou False caso contrário.

Considere a variável `sudokuBoard` como a matriz de teste do seu programa.

```
sudokuBoard = [  
    [5, 3, 4, 6, 7, 8, 9, 1, 2],  
    [6, 7, 2, 1, 9, 5, 3, 4, 8],  
    [1, 9, 8, 3, 4, 2, 5, 6, 7],  
    [8, 5, 9, 7, 6, 1, 4, 2, 3],  
    [4, 2, 6, 8, 5, 3, 7, 9, 1],  
    [7, 1, 3, 9, 2, 4, 8, 5, 6],  
    [9, 6, 1, 5, 3, 7, 2, 8, 4],  
    [2, 8, 7, 4, 1, 9, 6, 3, 5],  
    [3, 4, 5, 2, 8, 6, 1, 7, 9]  
]
```

## 2. Cálculo Recursivo do Fatorial

Crie uma função recursiva chamada ***calculateFactorial*** que receba como argumento um número inteiro positivo  $n$  e devolva o fatorial de  $n$ , definido como:

$$n! = n \times (n-1) \times (n-2) \times \dots \times 1$$

Com a condição especial de que  $0!=1$ .

Escreva um programa que permita testar a função recursiva criada.

### 3. Gestão de Livros

Crie uma classe chamada **Livro** que represente um livro. A classe deve ter os seguintes atributos e métodos:

Atributos:

titulo: O título do livro.

autor: O autor do livro.

ano\_publicacao: O ano em que o livro foi publicado.

numero\_paginas: O número de páginas do livro.

Métodos:

`__init__(self, titulo, autor, ano_publicacao, numero_paginas):`

O construtor da classe que inicializa os atributos titulo, autor, ano\_publicacao e numero\_paginas.

`atualizar_paginas(self, novas_paginas):`

Um método que recebe o número atualizado de páginas (por exemplo, para edições revistas) e atualiza o atributo numero\_paginas do livro.

`apresentar_informacoes(self):`

Um método que exibe as informações do livro, incluindo o título, autor, ano de publicação e número de páginas.

- a) Crie um objeto da classe Livro com informações sobre um livro à sua escolha.
- b) Atualize o número de páginas do livro utilizando o método *atualizar\_paginas*.
- c) Exiba as informações atualizadas do livro.

### 4. Calculadora

Crie um módulo chamado *calculadora* que contenha funções para realizar operações simples de adição, subtração, multiplicação e divisão. Em seguida, crie um programa principal que utilize esse módulo para realizar cálculos com base nas operações escolhidas pelo utilizador.

## 5. Contador de caracteres

Escreva um programa que calcule o número de ocorrências de cada caracter num texto.

- a) [1v] Crie uma função com o nome *readTextFile* que recebe como argumento o caminho para um ficheiro de texto "*lorem\_ipsum.txt*" que contem o texto a ser analisado. Esta função deve devolver uma string com o conteúdo do ficheiro.
- b) [2v] Crie uma função com o nome *countCharacters* que recebe como argumento um dicionário com todos os caracteres e os seus contadores e a string com o texto a analisar.
- c) [1v] Crie uma função com o nome *printCountCharacters* que recebe como argumento um dicionário com todos os caracteres e os seus contadores e imprime todas as ocorrências de cada caracter.
- d) [1v] Faça o tratamento da exceção caso o ficheiro não exista.

Nota: Caso não consiga resolver a alínea a) continue a resolução do exercício usando a variável seguinte:

loremIpsumText = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed in tincidunt diam. Mauris nec venenatis purus, eu sollicitudin nisi. In consequat volutpat malesuada. Morbi neque arcu, imperdiet hendrerit lacus sed, imperdiet fringilla quam. Ut ac tortor et nibh vehicula maximus eu at justo. Suspendisse potenti. Vivamus blandit ullamcorper luctus. Morbi tincidunt et purus et lacinia. Etiam lobortis congue lacus at tempus. Mauris mattis, felis id placerat faucibus, nisl ex viverra nibh, eget consequat ligula ipsum in diam. Nam id aliquam felis, id posuere quam. Mauris suscipit tincidunt bibendum. Suspendisse varius urna at neque dapibus, vitae ullamcorper quam cursus. Suspendisse accumsan velit erat, sit amet ultricies ipsum mattis in. Sed dapibus commodo est, sed commodo nulla egestas egestas. Aenean aliquam, velit sed ornare auctor, odio turpis laoreet sem, ut ultrices velit mauris ut nibh. Curabitur ut ligula feugiat, interdum neque nec, aliquam nulla. Morbi pharetra enim ipsum, eget interdum sem condimentum imperdiet. Praesent malesuada sem et ullamcorper condimentum. Integer vehicula lacinia cursus. Vestibulum ut tincidunt nibh, non gravida magna. Donec ligula neque, aliquam quis quam sed, tincidunt suscipit metus. Aliquam sapien nulla, porta sed lectus et, congue vulputate justo. Vivamus at placerat massa, vel pulvinar felis. Etiam pellentesque odio nec tincidunt."

## 1. Validador de Sudoku

```
def isValidGroup(group):
    group_sorted = sorted(group)
    return group_sorted == [1, 2, 3, 4, 5, 6, 7, 8, 9]

def extractColumn(board, col_index):
    column = []
    for row in board:
        column.append(row[col_index])
    return column

def isValidSudoku(board):
    # Verificar todas as linhas
    for row in board:
        if not isValidGroup(row):
            return False

    # Verificar todas as colunas
    for col_index in range(9):
        column = extractColumn(board, col_index)
        if not isValidGroup(column):
            return False

    return True

sudokuBoard = [
    [5, 3, 4, 6, 7, 8, 9, 1, 2],
    [6, 7, 2, 1, 9, 5, 3, 4, 8],
    [1, 9, 8, 3, 4, 2, 5, 6, 7],
    [8, 5, 9, 7, 6, 1, 4, 2, 3],
    [4, 2, 6, 8, 5, 3, 7, 9, 1],
    [7, 1, 3, 9, 2, 4, 8, 5, 6],
    [9, 6, 1, 5, 3, 7, 2, 8, 4],
    [2, 8, 7, 4, 1, 9, 6, 3, 5],
    [3, 4, 5, 2, 8, 6, 1, 7, 9]
]

# Verificar se o Sudoku é válido
if isValidSudoku(sudokuBoard):
    print("O Sudoku é válido!")
else:
    print("O Sudoku é inválido.")
```

## 2. Cálculo Recursivo do Fatorial

```
def calculateFactorial(n):
    if n == 0 or n == 1: # Caso base: 0! = 1 e 1! = 1
        return 1
    else:
        return n * calculateFactorial(n - 1)

print("Fatorial de 5:", calculateFactorial(5)) # Deve imprimir 120
print("Fatorial de 0:", calculateFactorial(0)) # Deve imprimir 1
print("Fatorial de 7:", calculateFactorial(7)) # Deve imprimir 5040
```

## 3. Gestão de Livros

```
class Livro:
    def __init__(self, titulo, autor, ano_publicacao, numero_paginas):
        self.titulo = titulo
        self.autor = autor
        self.ano_publicacao = ano_publicacao
        self.numero_paginas = numero_paginas

    def atualizar_paginas(self, novas_paginas):
        self.numero_paginas = novas_paginas

    def apresentar_informacoes(self):
        print(f"Título: {self.titulo}")
        print(f"Autor: {self.autor}")
        print(f"Ano de Publicação: {self.ano_publicacao}")
        print(f"Número de Páginas: {self.numero_paginas}")

# Criar um objeto da classe Livro
meu_livro = Livro("1984", "George Orwell", 1949, 328)

# Apresentar as informações iniciais do livro
print("Informações Iniciais do Livro:")
meu_livro.apresentar_informacoes()

# Atualizar o número de páginas do livro
meu_livro.atualizar_paginas(350)

# Apresentar as informações atualizadas do livro
print("\nInformações Atualizadas do Livro:")
meu_livro.apresentar_informacoes()
```

#### 4. Calculadora

# calculadora.py

```
def adicao(a, b):  
    return a + b
```

```
def subtracao(a, b):  
    return a - b
```

```
def multiplicacao(a, b):  
    return a * b
```

```
def divisao(a, b):  
    if b != 0:  
        return a / b  
    else:  
        return "Erro: Divisão por zero não é permitida."
```

# main.py  
import calculadora

```
def menu():  
    print("Escolha a operação:")  
    print("1 - Adição")  
    print("2 - Subtração")  
    print("3 - Multiplicação")  
    print("4 - Divisão")  
    print("5 - Sair")
```

```
def obter_operacao():  
    while True:  
        escolha = input("Digite o número correspondente à operação desejada: ")  
        if escolha in ['1', '2', '3', '4', '5']:  
            return int(escolha)  
        else:  
            print("Opção inválida. Tente novamente.")
```

```
def main():  
    while True:  
        menu()  
        escolha = obter_operacao()  
  
        if escolha == 5:  
            print("Sair do programa.")  
            break  
  
        a = float(input("Digite o primeiro número: "))
```

```

b = float(input("Digite o segundo número: "))

if escolha == 1:
    resultado = calculadora.adicao(a, b)
elif escolha == 2:
    resultado = calculadora.subtracao(a, b)
elif escolha == 3:
    resultado = calculadora.multiplicacao(a, b)
elif escolha == 4:
    resultado = calculadora.divisao(a, b)

print(f"O resultado é: {resultado}\n")

```

## 5. Contador de caracteres

```

def readTextFile(file_path):
    try:
        with open(file_path, 'r') as file:
            content = file.read()
            return content
    except FileNotFoundError:
        return None

def countCharacters(char_count_dict, text):
    for char in text:
        if char.isalnum(): # Conta apenas caracteres alfanuméricos (letras e números)
            if char in char_count_dict:
                char_count_dict[char] += 1
            else:
                char_count_dict[char] = 1

def printCountCharacters(char_count_dict):
    sorted_chars = sorted(char_count_dict.items()) # Ordena os caracteres
    alfabeticamente
    for item in sorted_chars:
        char = item[0] # Obtém a chave
        count = item[1] # Obtém o valor
        print(f"'{char}': {count}")

```

loremIpsumText = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed in tincidunt diam. Mauris nec venenatis purus, eu sollicitudin nisi. In consequat volutpat malesuada. Morbi neque arcu, imperdiet hendrerit lacus sed, imperdiet fringilla quam. Ut ac tortor et nibh vehicula maximus eu at justo. Suspendisse potenti. Vivamus blandit ullamcorper luctus. Morbi tincidunt et purus et lacinia. Etiam lobortis congue lacus at tempus. Mauris mattis, felis id placerat faucibus, nisl ex viverra nibh, eget consequat ligula ipsum in diam. Nam id aliquam felis, id posuere quam. Mauris suscipit tincidunt bibendum. Suspendisse varius urna at neque dapibus, vitae ullamcorper quam cursus. Suspendisse accumsan velit erat, sit amet ultricies ipsum mattis in. Sed

dapibus commodo est, sed commodo nulla egestas egestas. Aenean aliquam, velit sed ornare auctor, odio turpis laoreet sem, ut ultrices velit mauris ut nibh. Curabitur ut ligula feugiat, interdum neque nec, aliquam nulla. Morbi pharetra enim ipsum, eget interdum sem condimentum imperdiet. Praesent malesuada sem et ullamcorper condimentum. Integer vehicula lacinia cursus. Vestibulum ut tincidunt nibh, non gravida magna. Donec ligula neque, aliquam quis quam sed, tincidunt suscipit metus. Aliquam sapien nulla, porta sed lectus et, congue vulputate justo. Vivamus at placerat massa, vel pulvinar felis. Etiam pellentesque odio nec tincidunt."

```
file_path = "lorem_ipsum.txt"
```

```
try:
```

```
    text = readTextFile(file_path)
```

```
    if not text or text.strip() == "":
```

```
        text = loremIpsumText # conteúdo pré-definido caso o ficheiro não exista
```

```
    else:
```

```
        char_count = {}
```

```
        countCharacters(char_count, text)
```

```
        printCountCharacters(char_count)
```

```
except FileNotFoundError:
```

```
    print("Erro: O ficheiro não foi encontrado.")
```

```
except Exception:
```

```
    print("Erro inesperado.")
```