

**Objetivos:**

Utilizar boas práticas na resolução de problemas que envolvem conceitos de funções, listas, tuplas, dicionários e módulos.

**Tarefa:**

- Baixar o arquivo **ac01.py**.
- Inserir os nomes dos alunos do grupo no comentário inicial do arquivo.
- Implementar as funções definidas no arquivo de acordo com os comentários e descrições apresentadas abaixo (não esqueça que as funções devem retornar o resultado, utilizando return).
- Utilizar o arquivo de testes para auxiliar na verificação das funções implementadas.

**Atenção:**

- Não serão aceitos trabalhos entregues em atraso.
- Os nomes das funções contidas no arquivo **ac01.py** não devem ser alterados. Caso algum nome seja alterado, a função será considerada incorreta.
- Caso seja identificada a cópia de trabalhos entre os grupos, a nota será zerada.

**Forma de Entrega:**

- Anexar no Classroom o arquivo **ac01.py** com a implementação das funções solicitadas.
- Pode ser enviado por apenas um integrante do grupo (mas não esqueça de colocar o nome de todos os integrantes do grupo no início do arquivo).

**Descrição das Funções:**

1. Implemente a função '**posicoes**' que recebe como argumentos de entrada uma tupla e um item, e retorna uma lista contendo todos os índices em que o item aparece na tupla. Caso o item não exista na tupla, deve retornar uma lista vazia.

Exemplo:

```
tupla = (2, 1, 2, 3, 2, 2, 9, 2)
resultado = posicoes(tupla, 2)
print(resultado)                # [0, 2, 4, 5, 7]
```

2. Implemente a função '**remover\_itens\_repetidos**' que recebe como argumento de entrada uma lista e remove todos os itens repetidos dessa lista. A função deve retornar uma nova lista sem itens repetidos e ordenada de forma crescente.

Exemplo:

```
lista = [2, 1, 2, 3, 2, 2, 9, 2, 4, 5, 6, 6, 10, 9, 1, 1]
resultado = remover_itens_repetidos(lista)
print(resultado)                # [1, 2, 3, 4, 5, 6, 9, 10]
```

3. Considere um dicionário onde a chave é o nome de um aluno e o valor uma lista de notas. Implemente a função **'media\_notas'** que recebe como argumento de entrada o dicionário e retorna outro dicionário contendo as médias aritméticas de cada aluno.

Exemplo:

```
alunos = {'Augusto': [4.0, 7.0, 6.0, 3.0],
          'Denise': [9.0, 9.0, 9.0],
          'Ana Paula': [3.5, 9.5, 6.5],
          'Marcelo': [9.0, 10.0, 7.0, 7.0]}
resultado = media_notas(alunos)
print(resultado)                # {'Augusto': 5.0,
                                #  'Denise': 9.0,
                                #  'Ana Paula': 6.5,
                                #  'Marcelo': 8.25}
```

4. Considere um dicionário onde a chave é o nome de um aluno e o valor uma lista de notas. Implemente a função **'excluir\_menor\_nota'** que recebe como argumentos de entrada o dicionário e o nome de um aluno. A função deve excluir a menor nota do aluno informado e retornar o dicionário com as alterações realizadas.

Se aluno não estiver no dicionário, deve retornar o dicionário sem alterações.

Exemplo:

```
alunos = {'Augusto': [4.5, 7.0, 6.0, 3.0],
          'Denise': [9.0, 8.5],
          'Ana Paula': [3.5, 1.0, 6.5],
          'Marcelo': [9.0, 10.0, 7.0, 7.0]}
resultado = excluir_menor_nota(alunos, 'Denise')
print(resultado)                # {'Augusto': [4.5, 7.0, 6.0, 3.0],
                                #  'Denise': [9.0],
                                #  'Ana Paula': [3.5, 1.0, 6.5],
                                #  'Marcelo': [9.0, 10.0, 7.0, 7.0]}
```

5. Implemente a função **'caracter\_mais\_frequente'** que recebe como argumento de entrada uma string e retorna o caracter que aparece com mais frequência na string. Procure utilizar um dicionário para facilitar a implementação.

Exemplo:

```
texto = 'faculdade de tecnologia impacta'
resultado = caracter_mais_frequente(texto)
print(resultado)                # a

texto = 'exemplo de texto aleatorio para teste'
resultado = caracter_mais_frequente(texto)
print(resultado)                # e
```