Fatec
Rio Claro

Você recebeu o material necessário para realizar a primeira avaliação. São quatro atividades práticas que devem ser desenvolvidas no computador. Enviar na atividade do Teams somente os quatro arquivos contendo os códigos fontes. Não altere o nome do arquivo. A pontuação de cada atividade está na frente do enunciado. Manter os arquivos binários (*.bin) no mesmo diretório do código fonte.

Atividade 01:

Código: **exercicio01.py**

(1 ponto) Você foi contratado para auxiliar na organização de dados de uma empresa que realiza eventos voltados para diferentes faixas etárias. A empresa possui uma lista de participantes contendo nome e idade de cada pessoa, armazenada como uma lista de tuplas (**lista_carregada**)
Sua tarefa é implementar uma função chamada **menores_idade**, que receba uma lista de tuplas no formato [(nome, idade), ...] e retorne uma nova lista contendo somente os participantes com menos de 18 anos.

Entrada:

```
[("Ana", 17), ("Bruno", 18), ("Carlos", 16)]
```

Saída:

```
[("Ana", 17), ("Carlos", 16)]
```

Atividade 02:

Código: **exercicio02.py**

(1 ponto) Implemente a função remover_duplicados(lista_carregada) que:
Recebe como entrada uma lista de strings (lista_carregada), contendo nomes que podem estar repetidos. Retorna uma nova lista contendo apenas nomes únicos (sem duplicações). A ordem dos elementos na lista final não precisa ser preservada.

Entrada:

```
["Ana", "Bruno", "Ana", "Carla", "Bruno"]
```

Saída:

```
["Ana", "Bruno", "Carla"]
```

(1 ponto) Implemente a função numero_ocorrencias(lista_carregada) que:
Crie um dicionário onde cada palavra é uma chave e o valor é o número de vezes que ela aparece no texto.Retornar o dicionário.

Entrada:

```
["Ana", "Bruno", "Ana", "Carla", "Bruno"]
```

Saída:

```
{'Ana': 2, 'Bruno': 2, 'Carla': 1}
```

Fatec
Rio Claro

Atividade 03:

Código: **exercicio03.py**

Você recebeu um arquivo contendo uma lista de tuplas. Cada tupla representa uma pessoa e contém três informações:

- Nome da pessoa (string)
- Cidade onde mora (string)
- Um número decimal associado à pessoa (float entre 0 e 100)

Exemplo de tupla:

```
("Ana", "Araras", 87.5)
```

Implemente as seguintes funções para realizar buscas na lista carregada:

1.**(1 ponto)**
buscar_pessoa_por_cidade(lista_carregada, cidade)
Deve retornar uma nova lista contendo apenas as pessoas que moram na cidade informada.

A busca deve ser insensível a maiúsculas e minúsculas.
Exemplo: “Rio Claro” ou “rio claro” ou “RiO ClArO” serão tratadas como a mesma cidade.

2. **(1 ponto)**
buscar_pessoa_por_numero(lista_carregada, numero)
Deve retornar uma nova lista com todas as pessoas cujo número seja menor ou igual ao valor informado.

3. **(1 ponto)**
buscar_pessoa_por_numero_e_cidade(lista_carregada, cidade, numero)
Deve retornar uma nova lista com todas as pessoas que moram na cidade informada e possuem o número menor ou igual ao valor passado.

Atividade 04:

Código: **exercicio04.py**

Você recebeu um arquivo contendo uma lista de tuplas.
Cada tupla da lista possui o seguinte formato:
(nome, nota1, nota2, nota3, faltas).

(1 ponto) Implemente a função chamada **media_nota** que receba como parâmetro uma lista de tuplas contendo informações de alunos.
A função deve retornar uma nova lista de tuplas, onde cada tupla contenha o nome do aluno e a média aritmética das três notas (com duas casas decimais).

Exemplo de saída:

```
[('Ana', 7.5), ('Bruno', 6.17), ...]
```

(1 ponto) Implemente a função **situacao_falta** que recebe uma lista de tuplas com informações de alunos.
A função deve retornar uma nova lista de tuplas, onde cada tupla contenha o nome do aluno e a situação baseada na quantidade de faltas:

Se o aluno tiver 6 faltas ou mais, deve ser considerado "Reprovado".
Caso contrário, o aluno deve ser considerado "Aprovado".

Exemplo de saída:

```
[('Ana', 'Aprovado'), ('Bruno', 'Reprovado'), ...]
```

(1 ponto) Implemente a função **acima_de_nove** que receba uma lista de tuplas com as informações de alunos.
A função deve retornar uma lista contendo apenas os nomes dos alunos que obtiveram nota 9 ou mais em todas as três provas.

Exemplo de saída:

```
['Carlos', 'Kelly', 'Valéria']
```

(1 ponto) Implemente a função **uma_nota_abaixo_de_seis** que receba uma lista de tuplas com as informações de alunos.
A função deve retornar uma lista de tuplas, onde cada tupla contenha os dados de alunos que obtiveram pelo menos uma nota abaixo de 6.0.

Exemplo de saída:

```
[('Bruno', 6.0, 5.5, 7.0), ('Daniela', 4.5, 6.0, 5.8), ...]
```