

Sétima Lista de Exercícios - Computational Thinking

1. Crie uma tupla de números inteiros positivos com valores que você quiser, no mínimo 15 valores.

```
1 lista = (23, 45, 90, -10, 54, 87, -23, -1, 5, -17, 38, -26, 54, 90, -65, 28, 43, 76)
```

2. Imprima a tupla todos os números da tupla anterior na tela, um de cada vez.

```
1 lista = (23, 45, 90, -10, 54, 87, -23, -1, 5, -17, 38, -26, 54, 90, -65, 28, 43, 76)
2 i = 0
3 while i < len(lista):
4     print(lista[i])
5     i = i + 1
```

3. Faça uma função que recebe uma tupla de números inteiros e retorna o maior valor contido dentro da tupla.

```
1 def maior(tupla):
2     aux = tupla[0]
3     i = 1
4     while i < len(tupla):
5         if tupla[i] > aux:
6             aux = tupla[i]
7         i = i + 1
8     return aux
```

4. Faça uma função que recebe uma tupla de números reais e retorna a média aritmética de todos os valores contidos dentro da tupla.

```
1 def media(tupla):
2     soma = 0
3     i = 0
4     while i < len(tupla):
5         soma = soma + tupla[i]
6         i = i + 1
7     if i != 0:
8         return soma / i
9     else:
10        return 0
```

5. Faça uma função que recebe uma tupla de Strings e retorna a maior String de todos os valores contidos dentro da tupla.

```
1 def maiorString(tupla):
2     aux = len(tupla[0])
3     i = 1
4     while i < len(tupla):
5         if len(tupla[i]) > aux:
6             aux = len(tupla[i])
7         i = i + 1
8     return aux
```

6. O desvioPadrao é uma medida estatística que é representada pela letra grega σ . Ele pode ser calculado através da seguinte fórmula:

$$\sigma = \sqrt{\frac{\sum_{i=0}^{n-1} (x[i] - \bar{x})^2}{n - 1}} \quad (1)$$

onde \bar{x} é a média aritmética de todos os valores. Escreva uma função em Python que dada uma tupla de números reais, calcula o desvio padrão desses números.

```

1 def desvioPadrao(tupla):
2     #chamando a funcao media do exercicio anterior
3     med = media(tupla)
4     i = 0
5     soma = 0
6     while i < len(tupla):
7         soma = soma + (tupla[i] - med) * (tupla[i] - med)
8         i = i + 1
9
10    return soma / (len(tupla) - 1)

```

7. Dada uma tupla contendo nomes de Pessoas, escreva um programa que imprime todas as possíveis duplas que podemos formar com os nomes indicados, por exemplo:

```
1 nomes = ("Ana", "Bia", "Celi", "Diana", "Eva", "Fabia")
```

Seu programa deverá imprimir: Ana e Bia, Ana e Celi, Ana e Diana, Ana e Eva, Ana e Fabia, Bia e Celi,

Sugestão, crie uma função que recebe um nome, uma tupla e talvez mais um parâmetro. Sua função imprime na tela todas as possíveis duplas que posso formar com o nome que foi passado como parâmetro.

```

1
2 def mostraDuplas(lista, nome, pos):
3     while pos < len(lista):
4         print(nome, ' ', lista[pos])
5         pos = pos + 1
6
7 nomes = ("Ana", "Bia", "Celi", "Diana", "Eva", "Fabia")
8 for i in range(nomes):
9     mostraDuplas(nomes, nomes[i], i + 1)

```

8. Usando tuplas, crie todas as pedras que compõe um jogo de dominó, sugestão, faça uma tupla de tuplas. Como foi feito no Baralho.

```

1 domino = ((0,0), (0, 1), (0,2), (0, 3), (0, 4), (0, 5), (0, 6),
2            (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6),
3            (2, 2), (2, 3), (2, 4), (2, 5), (2, 6),
4            (3, 3), (3, 4), (3, 5), (3, 6),
5            (4, 4), (4, 5), (4, 6),
6            (5, 5), (5, 6),
7            (6, 6))

```

9. Segue abaixo a tabela que indica o cálculo do INSS.

| salário contribuição | alíquota/valor |
|----------------------------------|------------------------|
| até R\$ 1.317,07 | 8% |
| de R\$ 1.317,08 até R\$ 2.195,12 | 9% |
| de R\$ 2.195,13 até R\$ 4.390,24 | 11% |
| acima de R\$ 4.390,24 | 11% sobre R\$ 4.390,24 |

Tabela 1: Contribuição INSS 2014

Sua tarefa será a de criar uma tupla de tuplas que representa esta tabela. Sugestão, comece criando uma única tupla que representa uma linha da tabela.

```
1  tabela = (  
2      (1, 0, 1317.07, 0.08),  
3      (2, 1317.08, 2195.12, 0.09),  
4      (3, 2195.13, 4390.24, 0.11),  
5      (4, 4390.24, 10000000, 0.11 * 4390.24)  
6  )
```

Boa sorte!