

Capítulo 5

Listas

1. Escreva uma função que recebe uma cadeia de caracteres e que devolve a lista contendo os códigos “Unicode” de cada um dos caracteres da lista. Por exemplo:

```
>>> lista_codigos('bom dia')  
[98, 111, 109, 32, 100, 105, 97]
```

2. Escreva a função `remove_multiplos` que recebe uma lista de números e um número e que devolve a lista original em que todos os múltiplos do segundo argumento são removidos. Por exemplo:

```
>>> remove_multiplos([2, 3, 5, 9, 12, 33, 34, 45], 3)  
[2, 5, 34]
```

3. Escreva uma função em Python com o nome `junta_ordenadas` que recebe como argumentos duas listas ordenadas por ordem crescente e devolve uma lista também ordenada com os elementos das duas listas. Não é necessário validar os argumentos da sua função. Por exemplo,

```
junta_ordenadas([2, 5, 90], [3, 5, 6, 12])  
[2, 3, 5, 5, 6, 12, 90]
```

4. Escreva uma função, `soma_cumulativa`, que recebe uma lista de números e que devolve uma lista que contém a soma cumulativa da lista recebida, ou seja, o elemento na posição `i` da lista devolvida contém a soma de todos os elementos da lista original nas posições de 0 a `i`. Não é necessário validar os dados de entrada. Por exemplo,

```
>>> soma_cumulativa([1, 2, 3, 4, 5])  
[1, 3, 6, 10, 15]
```

5. A função de ordenação por borbulhamento apresentada na aula teórica corresponde a uma ordenação destrutiva, pois a lista original é destruída e substituída pela lista ordenada. Um processo alternativo consiste em criar uma lista com índices, que representam as posições ordenadas dos elementos da lista. Escreva uma função em Python para efetuar a ordenação por borbulhamento, criando uma lista com índices. A sua função deve permitir a interação:

```
>>> original = [5, 2, 9, 1, 4]
>>> ord = ordena(original)
>>> ord
[3, 1, 4, 0, 2]
>>> original
[5, 2, 9, 1, 4]
>>> for i in range(len(original)):
...     print(original[ord[i]])
...
1
2
4
5
9
```

6. Uma *matriz* é uma tabela bidimensional em que os seus elementos são referenciados pela linha e pela coluna em que se encontram. Uma matriz pode ser representada como uma lista cujos elementos são listas. Com base nesta representação, escreva uma função, chamada `elemento_matriz` que recebe como argumentos uma matriz, uma linha e uma coluna e que devolve o elemento da matriz que se encontra na linha e coluna indicadas. A sua função deve permitir a seguinte interação:

```
>>> m = [[1, 2, 3], [4, 5, 6]]
>>> elemento_matriz(m, 0, 0)
1
>>> elemento_matriz(m, 0, 3)
Índice inválido: coluna 3
>>> elemento_matriz(m, 4, 1)
Índice inválido: linha 4
```

7. Considere uma matriz como definida no exercício anterior. Escreva uma função em Python que recebe uma matriz e que a escreve sob a forma

$$\begin{array}{cccc}
 a_{11} & a_{12} & \cdots & a_{1n} \\
 a_{21} & a_{22} & \cdots & a_{2n} \\
 \cdots & \cdots & & \cdots \\
 a_{n1} & a_{n2} & \cdots & a_{nn}
 \end{array}$$

8. Considere, de novo, o conceito de matriz. Escreva uma função em Python que recebe como argumentos duas matrizes e devolve uma matriz correspondente à soma das matrizes que são seus argumentos. Sendo a e b as matrizes a somar, os elementos da matriz produto são dados por

$$s_{ij} = a_{ij} + b_{ij}$$

9. Considere, de novo, o conceito de matriz. Escreva uma função em Python que recebe como argumentos duas matrizes e devolve uma matriz correspondente ao produto das matrizes que são seus argumentos. Sendo a e b as matrizes a multiplicar, os elementos da matriz produto são dados por

$$p_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

10. A sequência de Racamán,

$$0, 1, 3, 6, 2, 7, 13, 20, 12, 21, 11, 22, 10, 23, 9, 24, \dots$$

é uma sequência de números inteiros não negativos, definida do seguinte modo: (1) o primeiro termo da sequência é zero; (2) para calcular o n -ésimo termo, verifica-se se o termo anterior é maior do que n e se o resultado de subtrair n ao termo anterior ainda não apareceu na sequência, neste caso o n -ésimo termo é dado pela subtração entre o $(n-1)$ -ésimo termo e n ; em caso contrário o n -ésimo termo é dado pela soma do $(n-1)$ -ésimo termo com n . Ou seja,

$$r(n) = \begin{cases} 0 & \text{se } n = 0 \\ r(n-1) - n & \text{se } r(n-1) > n \wedge (r(n-1) - n) \notin \{r(i) : 1 \leq i < n\} \\ r(n-1) + n & \text{em caso contrário} \end{cases}$$

Escreva uma função em Python que recebe um inteiro positivo, n , e devolve uma lista contendo os n primeiros elementos da sequência de Racamán. Por exemplo:

```
>>> seq_racaman(15)
[0, 1, 3, 6, 2, 7, 13, 20, 12, 21, 11, 22, 10, 23, 9]
```