

Computação I

Lista de exercícios 5 – A estrutura de repetição for e as listas de listas.

Atenção! Leia as instruções antes de fazer a lista! A padronização do nome do arquivo e dos nomes das funções é muito importante e está explicada no arquivo de instruções!

Data de entrega: 19/10/2022

A menos que esteja explicitamente pedido na questão, não utilize nenhum método ou função já existente do Python exceto pelas funções print, len, type e range (a função range só não pode ser usada nas questões 1, 2 e 3), e pelo método append. De forma simplificada, função é tudo o que precisa ser chamado com parênteses, e método é tudo o que precisa ser chamado com parênteses, mas que está associado a uma variável, isto é, variavel.metodo(args). Funções de transformação de tipo (int, float, str) também não são permitidas. Não importe nenhum módulo. Não é necessário testar se os dados passados por argumento são válidos. Nessa lista, utilize somente a estrutura de repetição for e **não utilize o while e não utilize recursão**. Não crie funções dentro de funções. Não use list comprehension.

1. Escreva uma função em Python que recebe por argumento uma lista que pode conter elementos do tipo inteiro, float, ou booleano, e retorna três listas, na seguinte ordem: a primeira contém somente os elementos do tipo inteiro da lista recebida por argumento, a segunda contém somente os elementos do tipo float da lista recebida por argumento, e a terceira contém somente os elementos do tipo booleano da lista recebida por argumento. A ordem dos elementos das listas criadas deve ser a mesma ordem que aparecia na lista original. Se não houver elementos de um determinado tipo, a lista retornada referente àquele tipo deve ser vazia. **Atenção! Para essa questão, não utilize a função range!**
2. Escreva uma função em Python que recebe, por argumento, uma lista de números e cria uma nova lista que contém os números da original, mas sem repetições. A nova lista deve ser retornada. Por exemplo, se a lista de entrada for ([1,2,2,0,1,3,9,8,0,7,4,0]), a lista retornada é ([1,2,0,3,9,8,7,4]) (mantenha a ordem original, somente retire as repetições). **Atenção! Para essa questão, não utilize a função range!**
3. Escreva uma função em Python que receba uma lista com números inteiros, em que cada elemento representa uma posição (índice) da própria lista passada como entrada. A função deve retornar uma lista na qual cada elemento corresponde a um elemento da lista original, porém ordenados segundo a indexação representada pelos elementos da lista de entrada. Por exemplo: para a entrada ([2,0,1]), a saída deve ser [1,2,0]. O primeiro elemento da lista de saída (1) corresponde ao elemento na posição 2 da lista de entrada. O segundo elemento da lista de saída (2) corresponde ao elemento na posição 0 da lista de entrada. O terceiro elemento da lista de saída (0) corresponde ao elemento na posição 1 da lista de entrada. Note que os elementos da lista de entrada podem ser negativos, e as mesmas regras de construção da lista de saída devem continuar valendo nestes casos. Por exemplo: para a entrada ([-2,-1,-3]), a saída deve ser ([-1,-3,-2]). O primeiro elemento da lista de saída (-1) corresponde ao elemento na posição -2 da lista de entrada. O segundo elemento da lista de saída (-3) corresponde ao elemento na posição -1 da lista de entrada. O terceiro elemento da lista de entrada (-2) corresponde ao elemento na posição -3 da lista de entrada.

Se um elemento da lista de entrada corresponder a um índice fora dos índices permitidos da própria lista, a lista de saída deverá conter o valor booleano "False" na posição em que tal elemento estava na lista de entrada. Por exemplo: para a entrada ([2,3,1]), a saída deve ser ([1,False,3]). Para a entrada ([-4,-1,-3]), a saída deve ser ([False, -3, -4]). **Atenção! Para essa questão, não utilize a função range!**

4. Escreva uma função em Python que receba uma lista de números e retorne uma lista em que cada posição contém o valor da média aritmética dos elementos na lista de entrada localizados na posição

imediatamente anterior e imediatamente posterior à posição em questão. Assuma que as posições extremas da lista de saída (inicial e final) contêm o valor do respectivo elemento adjacente na lista de entrada dividido por 2. Por exemplo: para a entrada $[4,3,9,1]$, a saída deve ser $[1.5,6.5,2.0,4.5]$. O primeiro elemento da lista de saída corresponde ao segundo elemento da lista de entrada dividido por 2, ao passo que o último elemento da lista de saída corresponde ao penúltimo elemento da lista de entrada dividido por 2. Os demais elementos da lista de saída correspondem às médias entre os elementos da lista de entrada que estão em posições adjacentes à posição em sendo considerada na lista de saída.

5. Faça uma função em Python que recebe 3 argumentos, nesta ordem: uma lista, um inteiro que vale 1 ou -1, e um inteiro positivo n . A função deve retornar uma lista contendo os elementos da lista de entrada deslocados n posições para a direita (se o segundo argumento for 1) ou para a esquerda (se o segundo argumento for -1). No caso de deslocamento para a direita, as primeiras n posições da lista de saída devem ser preenchidas com 0. No caso de deslocamento para a esquerda, as últimas n posições da lista devem ser preenchidas com 0. Por exemplo: para a entrada $[1,2,3,4]$, 1, 2), a saída deve ser $[0,0,1,2]$, pois todos os elementos foram deslocados duas posições para a direita, e as duas primeiras posições são preenchidas com 0 (observe que os dois últimos elementos da lista de entrada não estão mais presentes na lista de saída). Já para a entrada $[1,2,3,4]$, -1, 2), a saída deve ser $[3,4,0,0]$, pois todos os elementos foram deslocados duas posições para a esquerda, e as duas últimas posições são preenchidas com 0 (neste caso, os dois primeiros elementos da lista de entrada não estão mais presentes na lista de saída). Observe que a lista de saída deve sempre ter o mesmo tamanho da lista de entrada. **Atenção! Para essa questão, não utilize o operador de fatiamento de lista!**
6. Escreva uma função em Python que receba uma lista de listas, onde cada sublista contém 3 elementos, representando, cada um, as seguintes informações sobre uma pessoa (nesta ordem): sua idade (inteiro), sua altura (float) e o seu peso (float). A função deve retornar uma lista com 3 elementos, contendo, nesta ordem: a média das idades, a média das alturas e a média dos pesos, considerando todas as pessoas representadas pelas sublistas.
7. Faça uma função em Python que receba duas listas de números e retorna uma lista de listas, em que cada sublista contém como elementos os produtos entre um dado elemento da primeira lista e cada elemento da segunda lista. Por exemplo: para a entrada $[1,2,3],[4,5,6]$, a saída deve ser $[[4,5,6],[8,10,12],[12,15,18]]$. A segunda sublista na lista de listas de saída $[8,10,12]$ corresponde aos produtos envolvendo o primeiro elemento da primeira lista de entrada (2) e o todos os elementos da segunda lista de entrada (4, 5 e 6).
8. Em processamento de imagens, imagens são representadas como matrizes, e cada elemento da matriz, também chamado de "pixel", corresponde a um valor numérico. Uma operação muito comum aplicada após processar uma imagem consiste em limitar valores dos pixels dentro de um intervalo específico, como entre 0 e 255. Neste exemplo, pixels com valores negativos têm seus valores trocados para 0, enquanto que pixels com valores maiores do que 255 têm seus valores trocados para 255. Os demais pixels ficam com valores inalterados.

Inspirado nessa aplicação, faça uma função em Python que receba por argumento uma lista de listas representando uma matriz, e que retorne a mesma lista de listas, em que os valores estejam dentro do intervalo de 0 a 10. Por exemplo, para a entrada $[[-1,0,5,-4], [-3,8,12,20],[1,2,7,10]]$, a saída deve ser $[[0,0,5,0], [0,8,10,10], [1,2,7,10]]$. Note que os elementos que já possuem valores dentro da faixa permitida permanecem inalterados, e a ordem dos elementos nas listas permanece a mesma.