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: 01/06/2021

A menos que esteja explicitamente pedido na questão, não utilize nenhum método ou função já existente do Python exceto pelas funções print, len e range (a função range só não pode ser usada nas questões 1 e 7), 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 uma lista de números por argumento e retorna uma outra lista com todos os elementos da primeira multiplicados por 2. **Para essa questão, não utilize a função range.**

2. Faça uma função em Python que receba dois argumentos, nesta ordem: uma lista de números e um número n. A função deve retornar uma lista, de mesmo tamanho da lista de entrada, que substitui os elementos da lista original por 0 (se o valor original, na respectiva posição, for menor ou igual ao número n passado como segundo argumento da função), ou por 1 (se o valor original, na respectiva posição, for maior do que o número n passado como segundo argumento da função).

3. Faça uma função em Python que receba duas listas, ambas do mesmo tamanho, e retorne uma lista que intercala os elementos da primeira lista com os da segunda lista. Por exemplo: para a entrada ([3,4,5,6], [1,2,9,3]), a função deve retornar [3,1,4,2,5,9,6,3].

4. Escreva uma função em Python que receba uma lista de números e retorne o valor da soma dos elementos localizados em posições (índices) ímpares menos a soma dos elementos localizados em posições (índices) pares (o índice 0 é um índice par). Por exemplo: para a entrada [1,2,3,4,5], o retorno deve ser -3, pois (2+4) - (1+3+5) = 6 - 9 = -3. **Atenção! Não é permitido usar a função sum!**

5. Triângulos são polígonos que contêm três lados. Existem diversos tipos de triângulos que aparecem em muitos problemas matemáticos e de engenharia. Entretanto, há certas propriedades que todo o triângulo deve atender, independente do seu tipo. Uma delas é a de que cada lado do triângulo deve ser sempre menor do que a soma dos outros dois lados. Sua tarefa é escrever uma função em Python que ajude a descobrir se, dado 3 números positivos, eles podem representar os lados de um triângulo ou não. A função deve receber uma lista de listas, em que cada sublista contém 3 números positivos que serão analisados. A função deve retornar uma lista de booleanos, que indica se os valores da sublista localizada na respectiva posição da lista de entrada correspondem a um triângulo válido ou não. Por exemplo: para a entrada [[3,4,5],[10,15,30]] deve ser [True, False]. O valor booleano "True" ocorre na primeira posição da lista de saída, porque a sublista [3,4,5], que é o primeiro elemento da lista de entrada, atende à condição imposta (5 é menor do que 3 + 4, 4 é menor do que 5 + 3, e 3 é menor do que 5 + 4). Já o valor booleano "False" ocorre na segunda posição da lista de saída, porque a sublista [10,15,30], que é o segundo elemento da lista de entrada, não representa um triângulo válido, porque 30 é maior do que 10 + 15.

6. Escreva uma função em Python que recebe uma lista de listas que representa uma lista de produtos que serão comprados. Cada sublista contém dois números (inteiros ou floats) positivos: um peso, em

quilos, de um produto e o valor por quilo desse produto. A função deve retornar o preço total que deve ser pago. Veja que é necessário acessar cada sublista, pois o preço a ser pago por um produto é dado por peso*valor, onde peso é o elemento da posição 0 da sublista referente àquele produto e valor é o elemento da posição 1 da sublista referente àquele produto. O valor total a ser pago (retornado) é o somatório dos preços a serem pagos. Por exemplo, se a entrada for ([[1,2],[3,4],[5,6]]), o valor de saída deve ser 1*2+3*4+5*6 = 44.

7. O histograma é um gráfico em colunas, muito utilizado em análises estatísticas, que possui, no eixo x, um conjunto de classes em que os dados estão distribuídos, e, no eixo y, a quantidade de ocorrências de cada classe. Veja abaixo o exemplo de um histograma com 4 classes:

Esse histograma indica que, ao analisar os dados coletados, foram detectadas quatro ocorrências da classe 0, sete ocorrências da classe 1, quatro ocorrências da classe 2 e duas ocorrências da classe 3. Escreva uma função em Python que, dados os valores de um experimento com 20 classes, calcula o histograma referente a esse experimento. Sua função recebe por argumento uma lista de qualquer tamanho que contém números inteiros de 0 a 19, os quais representam os resultados coletados do experimento, e calcula a quantidade de vezes que cada número (entre 0 e 19) apareceu nessa lista. Sua função deve retornar uma lista com 20 posições contendo: na posição 0, a quantidade de aparições do número 0 na lista de entrada; na posição 1, a quantidade de aparições do número 1 na lista de entrada; na posição 2, a quantidade de aparições do número 2 na lista de entrada, e assim por diante. Ou seja, cada posição da lista retornada representa uma classe, que estaria no eixo x do histograma, e os elementos da lista retornada representam os valores do histograma no eixo y. **Atenção! Para essa questão, não utilize a função range.** Dica: veja que é possível criar uma lista de tamanho n vezes maior que uma lista de referência através de uma multiplicação da lista de referência por n. Com isso, é possível, por exemplo, criar uma lista com cinco posições e com todos os elementos iguais a zero com a seguinte atribuição: listaExemplo = [0]*5 . Após essa atribuição, a variável listaExemplo faz referência à lista [0,0,0,0,0].

8. Dizemos que uma sequência de números é uma escadinha, se a diferença entre números consecutivos é sempre a mesma. Por exemplo, “2, 3, 4, 5” e “10, 7, 4” são escadinhas. Note que qualquer sequência com apenas um ou dois números também é uma escadinha! Neste problema estamos procurando escadinhas em uma sequência maior de números. Dada uma sequência de números, queremos determinar quantas escadinhas existem. Mas só estamos interessados em escadinhas tão longas quanto possível. Por isso, se uma escadinha é um pedaço de outra, consideramos somente a maior. Por exemplo, na sequência “1, 1, 1, 3, 5, 4, 8, 12” temos 4 escadinhas diferentes: “1, 1, 1”, “1, 3, 5”, “5, 4” e “4, 8, 12”. Faça uma função em Python que recebe por argumento uma lista de números inteiros definindo a sequência e retorna um m inteiro representando quantas escadinhas existem na sequência. Tome cuidado com um detalhe: o último número de uma sequência é o primeiro da sequência seguinte (exceto pela última sequência).