

Computação I

Lista de exercícios 8 – Valores padrão para argumentos e tratamento de exceção

Atenção! Leia as instruções antes de fazer a lista! - Data de entrega: 06/07/2022

Não importe nenhum módulo. Não é necessário testar se os dados passados por argumento são válidos a menos que pedido na questão. Veja no enunciado da questão se há alguma restrição para a utilização de métodos e funções do Python. Se nada for dito no enunciado, eles estão liberados. Utilize tratamento de exceção nas questões 2, 3 e 4.

1. Faça uma função em Python que calcule uma determinada média de uma lista de números. A função recebe dois argumentos, nesta ordem: uma lista de inteiros ou floats positivos e uma string. A string especifica o tipo de média a ser calculada: 'A' para média aritmética, 'G' para média geométrica, ou 'H' para média harmônica. As médias geométrica G e harmônica H de um conjunto numérico (x_1, x_2, \dots, x_n) são dadas por:

$$G = \sqrt[n]{x_1 \cdot x_2 \cdot \dots \cdot x_n} \quad \text{e} \quad H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

A função deve retornar a média dos números da lista, de acordo com o que foi especificado pela string. Se a string for diferente de todas as opções especificadas, retorne -1. **O segundo argumento da sua função deve ter valor padrão igual a 'A'**. Assuma que os dados de entrada sempre serão válidos (isto é, os números sempre serão inteiros ou floats positivos). Não utilize a função "sum()" do Python.

2. Faça uma função em Python que recebe duas listas por argumento e retorna uma nova lista que contém as médias, elemento a elemento dos elementos das listas recebidas por argumento. Sua função deve percorrer as listas elemento a elemento e **usar tratamento de exceção para tentar calcular a média dos elementos das mesmas posições. Utilize uma única estrutura de tratamento de exceções (apenas um 'try' e múltiplos 'except')**. Se nenhuma exceção for gerada, a média deve ser incluída na lista que será retornada. Se os elementos não puderem ser somados, a exceção `TypeError` será gerada e você deve encerrar a função retornando o inteiro -1. Se as listas forem de tamanhos diferentes, a exceção `IndexError` poderá ser gerada ao percorrer as listas. Caso a exceção `IndexError` seja gerada, você deve interromper o laço de repetição e retornar a lista com as médias que foram calculadas até o momento. **Não utilize condicional para verificar o tipo dos elementos ou verificar se uma lista é maior que outra, pois o objetivo aqui é praticar o uso do tratamento de exceção.**
3. Faça uma função em Python que recebe uma string, que é o nome de um arquivo, por argumento. A função deve ler o arquivo de texto que contém, em cada linha, a temperatura de uma cidade ao longo de um dia; e deve retornar a temperatura máxima. **Utilize tratamento de exceção ao tentar abrir o arquivo para leitura.** Se não for possível abrir o arquivo para leitura, imprima uma mensagem de erro, indicando que o arquivo não existe, e retorne -1 (lembre-se: imprima com o `print`, retorne com o `return`). **Utilize tratamento de exceção também ao tentar converter cada linha do arquivo para float.** Além da temperatura, o arquivo pode conter linhas com outras informações, como uma informação de hora ou data, por exemplo. Se não for possível converter uma linha, imprima uma mensagem indicando que uma linha não pôde ser convertida, e informando a linha em si (veja o exemplo de execução no pdf com os testes), mas continue convertendo as demais linhas e utilizando as linhas que puderam ser convertidas para float na verificação do valor máximo. O valor de retorno deve ser do tipo float. Considere que as temperaturas do arquivo são, necessariamente, maiores que -1000. Para essa questão, não utilize as funções `max` e `min` do Python e não utilize o método `sort`.

4. Faça uma função em Python que ajuda a obter informações sobre um campeonato de futebol de 5 times. O programa deve interagir com o usuário, pedindo as seguintes informações, nesta ordem: nome do time (string), pontuação do time no campeonato (inteiro), e média de gols por jogo do time (float). As informações devem ser pedidas uma de cada vez, usando a função input. A função deve retornar um dicionário com as seguintes chaves: 'Campeao', 'Melhor media gols'. Os respectivos valores associados a cada chave devem ser os seguintes: uma string contendo o nome do time que alcançou a maior pontuação, e uma lista de strings contendo os nomes dos times com maior média de gols. Caso dois times tenham a mesma pontuação, o campeão será aquele que possuir a maior média de gols dentre os que empataram. Por simplicidade, assuma que nunca haverá mais de um time com a mesma pontuação e média de gols. Ao pedir a pontuação, **utilize tratamento de exceção para garantir que o valor digitado pelo usuário possa ser convertido para inteiro e que o valor está entre 0 e 24 (utilize o raise para definir os limites e levante a exceção ValueError se os limites forem violados). Ou seja, se a exceção ValueError for gerada, então o valor é inválido.** Caso o valor seja inválido, imprima a mensagem de erro "Pontuacao deve ser um inteiro entre 0 e 24" e repita a interação de pontuação até que o valor digitado seja válido. A média de gols só deve ser pedida quando o valor passado para a pontuação for válido. **Utilize tratamento de exceção novamente para pedir a média de gols do time. O programa deve identificar se o valor digitado pode ser convertido para float, e se ele é maior ou igual a 0 (de novo, utilize o raise para definir os limites e levante a exceção ValueError se o limite for violado).** Caso o valor seja inválido, imprima a mensagem de erro "Media de gols deve ser um float maior ou igual a 0", e repita a interação de média de gols até que o valor digitado seja válido. **Atenção! Utilize exatamente as mesmas mensagens, com os mesmos caracteres, para as strings que são impressas na tela (olhar arquivo de testes)! Isso facilita consideravelmente a correção.**