

Computação I

Lista de exercícios 3 – A estrutura de repetição while e chamada de função dentro de função.

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: 18/05/2021

A menos que esteja explicitamente pedido na questão, não utilize nenhum método ou função já existente do Python exceto pela função print e pela função abs (apenas na questão 3). 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 while e **não utilize o for e não utilize recursão**. Não crie funções dentro de funções.

1. A função exponencial natural pode ser calculada utilizando a seguinte série:

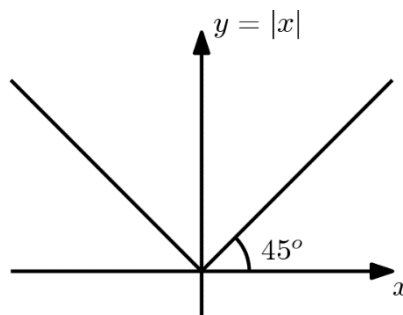
$$e^x = \sum_{n=0}^m \frac{x^n}{n!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \dots + \frac{x^m}{m!}$$

Veja que m representa o número de termos do somatório que são usados. Nessa questão você deve, necessariamente, fazer duas funções em Python: uma para cálculo do fatorial (cujo nome é livre); e outra para cálculo da função exponencial (cujo nome deve terminar com o número 1, por exemplo, questao1). A função para cálculo do fatorial recebe por argumento um número inteiro e maior ou igual a 0 e retorna o fatorial desse número (lembrando que o fatorial de 0 é 1). A função para cálculo da exponencial (aquela cujo nome termina com 1) recebe os números x e m por argumento (e nessa ordem) e calcula a função exponencial de acordo com o somatório acima. A função de cálculo da função exponencial deve chamar a função fatorial para encontrar dos denominadores. Considere que x é um float e que m é um inteiro positivo e maior ou igual a 0. A função questao1 deve retornar o resultado da exponencial de acordo com o cálculo da série.

2. Escreva uma função em Python que recebe quatro números inteiros positivos como argumentos. O primeiro e segundo argumentos representam, respectivamente, o valor mínimo e máximo de uma faixa de números que será analisada. O terceiro e quarto argumentos representam os múltiplos que queremos verificar dentro dessa faixa de valores. A função deve retornar três números inteiros na seguinte ordem: a quantidade de números dentro da faixa considerada que são múltiplos do terceiro argumento **mas não** do quarto argumento, a quantidade de números dentro da faixa considerada que são múltiplos do quarto argumento **mas não** do terceiro argumento, e a quantidade de números múltiplos **tanto** do terceiro **quanto** do quarto argumento. Por exemplo: para a entrada (3,8,2,3), a função deve retornar (2,1,1), pois, entre 3 e 8, existem 3 números múltiplos de 2 mas não de 3 (4 e 8), existe apenas 1 número múltiplo de 3 mas não de 2 (3), e existe apenas 1 número múltiplo de 2 e 3 (6). Veja o arquivo de testes para mais exemplos. **Atenção! Não é permitido usar o operador de divisão inteira (//) nesta questão! É obrigatório utilizar o loop while para resolver a questão!**
3. Considere um jogo em que uma pessoa deve programar uma máquina para arremessar um peso que deve atingir um alvo em uma determinada posição em menos de um número pré-definido de tentativas. A força do arremesso depende da distância atual que o peso se encontra da máquina, e de uma constante definida pela pessoa ao configurar a máquina. A fórmula que define a posição do peso, após um arremesso, é dada a seguir:

$$P = P_0 - f \times (P_0 - A)$$

Na fórmula acima, P é a posição final do peso após o arremesso, P_0 é a posição do peso logo antes de ser arremessado, A é a posição do alvo, e f é a constante configurada pela pessoa. Observe que se o alvo estiver à frente da posição atual ($A > P_0$), então a posição final estará também à frente da posição atual ($P > P_0$). Do contrário, se o alvo estiver atrás da posição atual ($A < P_0$), então a posição final também estará atrás da posição atual ($P < P_0$). Faça uma função em Python que receba números representando, nesta ordem, a posição inicial do peso (em metros), a posição do alvo (em metros), a constante f definida pela pessoa, e o número máximo de tentativas permitido. A função deve retornar dois valores, nesta ordem: um inteiro representando o número de tentativas de arremesso, e um booleano que indica se o peso atingiu o alvo (True) ou não (False), para tal número de tentativas. Assuma que o peso atingiu o alvo caso a distância entre eles seja menor do que 0,1 metros. Para essa questão, é permitido o uso da função `abs()` do Python para verificar se a distância entre o peso e o alvo atende à condição desejada. A função `abs(x)` recebe um valor numérico x e retorna o módulo $|x|$ desse número. Por exemplo: `abs(-2.5)` retorna 2.5; `abs(4.33)` retorna 4.33. A representação gráfica da função módulo é mostrada abaixo:



4. Em matemática, um número perfeito é um número inteiro para o qual a soma de todos os seus divisores positivos próprios (excluindo ele mesmo) é igual ao próprio número. Por exemplo, o número 6 é um número perfeito, pois: $6 = 1 + 2 + 3$. Faça uma função em Python que recebe um número n por argumento, inteiro e maior que zero, e retorna o booleano True se ele for perfeito ou o booleano False se não for.
5. Para essa questão, você deverá escrever, obrigatoriamente, duas funções em Python. A primeira função, cujo nome deve terminar com 5a, deve receber um argumento inteiro e retornar o quociente inteiro e o resto da divisão do argumento de entrada por 10 (exemplos de nomes de função válidos: `calcularQuocienteResto5a`, `quocienteResto5a`, `questao5a...`). A segunda função, cujo nome deve terminar com 5b (por exemplo: `contarOcorrencias5b`, `contarDigitos5b`, `questao5b`, ...), recebe dois argumentos inteiros não-nulos. O primeiro argumento representa uma sequência de dígitos de 0 a 9, enquanto que o segundo argumento representa um dígito apenas. Essa segunda função deve retornar a quantidade de vezes que o dígito passado no segundo argumento ocorre na sequência passada como primeiro argumento. Por exemplo: para a entrada (22391274, 2), a função cujo nome termina com 5b deve retornar 3, pois o dígito 2 aparece 3 vezes na sequência 22391274. Veja que para qualquer número $wxyz$ que for o argumento da função da questão 5a, os valores de retorno serão wxy e z , onde z é o dígito atual (que desejamos verificar se é igual ao dígito passado como segundo argumento da função da questão 5b), e wxy é o restante do número, que ainda precisa ser verificado. Por exemplo, se a entrada da questão 5a for 3093, os valores de retorno devem ser 309 e 3. Para pegar os dois valores retornados pela questão 5a separadamente, ao fazer a chamada da função 5a dentro da função 5b, faça uma atribuição a duas variáveis. Por exemplo, para um número n igual a 55942, faça a chamada da seguinte forma:
`quociente, resto = questao5a(n)`
Nesse caso, a variável `quociente` receberá o valor 5594 e a variável `resto` receberá o valor 2. As variáveis `quociente` e `resto` podem, portanto, ser usadas separadamente.