Computação I

Lista de exercícios 9 – Recursão
Atenção! Leia as instruções antes de fazer a lista! - Data de entrega: 13/07/2022
Não é necessário testar se os dados passados por argumento são válidos a menos que pedido na questão. **Não utilize nenhuma função ou método exceto pelas funções len e print, a menos que especificado no enunciado que é permitido usar outra função.** Os exercícios abaixo devem ser resolvidos, necessariamente, utilizando recursão: não utilize for e while nesses exercícios (exceto pelo exercício 6). Soluções sem recursão não serão aceitas (exceto pelo exercício 6). Não importe nenhum módulo. A questão 6 é opcional e vale quatro pontos extras: a nota máxima dessa lista é 14.

1. Faça uma função que retorna o máximo divisor comum (mdc) entre dois números inteiros. Os dois números inteiros são recebidos através de argumentos da função. Utilize recursão para resolver o problema sabendo que o mdc pode ser calculado da seguinte forma:
   mdc(x,x) = x;
   mdc(x,y) = mdc(x-y,y), se x > y;
   mdc(x,y) = mdc(x,y-x), se y > x .

2. Faça uma função recursiva que recebe uma lista de números por argumento e retorna a quantidade de vezes que um número negativo aparece na lista.

3. Em processamento de sinais, uma técnica muito utilizada para lidar com sinais ruidosos consiste em filtrá-los por meio de um algoritmo chamado de média móvel. O que esse algoritmo faz é calcular a média dos últimos M valores de um sinal de entrada, contados a partir de um determinado instante de tempo t (e incluindo o valor do sinal no instante de tempo em questão). O valor M caracteriza o tamanho do filtro de média móvel. Por exemplo: considere um sinal de entrada $x(t)$ com 5 valores medidos em cada segundo ($t = 0s, 1s, 2s, 3s, 4s$). Se $M = 3$, então a saída do filtro para o instante de tempo $t = 4s$ será $\frac{x(4) + x(3) + x(2)}{3}$. O filtro de média móvel com $M = 3$ pode ser implementado recursivamente da seguinte forma:

   $$y(t) = y(t - 1) - \frac{x(t-3)}{3} + \frac{x(t)}{3}$$

   Na equação acima, $y(t)$ é a saída do filtro no instante t, enquanto que $x(t)$ representa o valor do sinal de entrada no instante t. Faça uma função recursiva em Python que implemente o filtro média móvel acima. A função recebe uma lista de números que representa os valores de um sinal $x(t)$ lidos a cada segundo, e um inteiro maior ou igual a 0 que representa o instante de tempo t para o qual desejamos saber a saída do filtro. A função retorna o valor da saída do filtro para o instante t especificado. Na equação acima, considere que o sinal $x(t)$ vale 0 para todo $t < 0$ (ou seja, $x(-1) = x(-2) = x(-3) = 0$). Consequentemente, a saída $y(t)$ também será 0 para todo $t < 0$ (ou seja, $y(-1) = 0$). Observe que, em decorrência disso, a saída do filtro para $t = 0s$ será $x(0)/3$, ao passo que, para $t = 1s$, ela será $(x(0) + x(1))/3$. Veja o arquivo de testes para alguns exemplos.

4. Escreva uma função recursiva em Python que receba uma lista de valores 'L', e retorne a mesma lista com a ordem dos elementos invertida. Por exemplo: para a entrada [1,2,3,4,5], a saída deve ser [5,4,3,2,1]. Para essa questão, é permitido o uso do operador de fatiamento, porém **não é permitido inverter a lista pelo comando [::-1] (por exemplo, L[::-1]).**

5. Para essa questão, não utilize nenhum método e não utilize o operador in. Crie uma função em Python que implemente o algoritmo de busca binária de forma recursiva considerando que os argumentos da função

são: uma lista de números ordenada de forma decrescente e um número (inteiro ou float). A função deve retornar a posição da lista que contém o elemento mais à esquerda igual ao número passado por argumento, ou, caso não haja nenhum elemento igual ao número, a posição em que o número deve ser colocado, de forma a manter a ordem decrescente (isto é, a posição do primeiro elemento menor que o número). O algoritmo de busca binária foi implementado no vídeo 44. Para essa questão, vocês podem modificar o algoritmo fornecido ou criar o algoritmo desde o começo. A busca binária funciona da seguinte forma:

   a. Compare o número dado com o número da posição central da lista.
   b. Se o número for menor, repita o passo (a) para a primeira metade da lista.
   c. Se o número for maior, repita o passo (a) para a segunda metade da lista.
   d. Repita os passos até que a posição desejada tenha sido encontrada ou até que a região de busca se reduza a somente um elemento.

6. (Questão opcional – 4 pontos extras nesta lista – não utilize recursão nesta questão, tanto o for quando o while são permitidos, e a função range também pode ser usada) Implemente o algoritmo insertion sort (esse algoritmo foi explicado no vídeo 45).