Computação 1, 2021.2
Lista 5

Data limite para entrega: **18/1** às 18:00

Submeta suas soluções colocando os arquivos correspondentes na sua pasta do Google Drive*

Atualizada em 13/01, estendendo o prazo de entrega.

Parte 1 — Obrigatória

Questão 1. Faça uma função que receba como entrada 3 inteiros `termo_inicial`, `razão` e `num_termos` e retorne um `range` correspondente à P.A. que tem termo inicial `termo_inicial`, razão `razão` e exatamente `num_termos` termos.

Questão 2.

a. Em um alfabeto com apenas dois símbolos, `+` e `-`, digamos que uma expressão é *bem formada* se:

1. as quantidades de `+`s e `-`s na expressão são iguais;
2. à esquerda de cada símbolo `-`, a quantidade de `+`s é estritamente maior que a quantidade de `-`s.

Portanto `++-+--` é bem formada mas `-+--++` não é.

Faça uma função que receba uma string correspondendo a uma expressão neste alfabeto e retorne um booleano indicando se a expressão é bem formada ou não.

b. Em um alfabeto com apenas dois símbolos, `)` e `(`, digamos que uma expressão é *bem formada* se ela pode ser obtida de alguma expressão aritmética bem formada (de acordo com as regras aprendidas no ensino fundamental) apagando-se todos os símbolos que não são `)` nem `(`. Por exemplo, a expressão `(()(()))` é bem formada, pois poderia ter vindo da expressão aritmética `((2+3)/((1+1)*2))`, mas a expressão `(()` não é bem formada.

Faça uma função que receba uma string correspondendo a uma expressão neste alfabeto e retorne um booleano indicando se a expressão é bem formada ou não.

*Link recebido por email em 24/11/2021 — o nome é parecido com `<seu nome> - Comp 1 2021.2 - Submissões e Feedback`.

Questão 3 (Matrizes)**.** Podemos representar uma matriz real M com ℓ linhas e c colunas,

$$M = \begin{pmatrix} m_{0,0} & m_{0,1} & m_{0,2} & \cdots & m_{0,c-1} \\ m_{1,0} & m_{1,1} & m_{1,2} & \cdots & m_{1,c-1} \\ m_{2,0} & m_{2,1} & m_{2,2} & \cdots & m_{2,c-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{\ell-1,0} & m_{\ell-1,1} & m_{\ell-1,2} & \cdots & m_{\ell-1,c-1} \end{pmatrix},$$

em `python` como uma lista `L` de comprimento ℓ, onde cada elemento é por sua vez uma lista de comprimento c, de forma que dados $i < \ell$ e $j < c$ tenhamos

$$\mathtt{L}[i][j] = \tilde{m}_{i,j},$$

sendo $\tilde{m}_{i,j}$ o `float` correspondente ao real $m_{i,j}$.

Em outras palavras,

$$\begin{aligned} \mathtt{L} = [&[\tilde{m}_{0,0},\, \tilde{m}_{0,1},\, \tilde{m}_{0,2},\, \ldots,\, \tilde{m}_{0,c-1}], \\ &[\tilde{m}_{1,0},\, \tilde{m}_{1,1},\, \tilde{m}_{1,2},\, \ldots,\, \tilde{m}_{1,c-1}], \\ &[\tilde{m}_{2,0},\, \tilde{m}_{2,1},\, \tilde{m}_{2,2},\, \ldots,\, \tilde{m}_{2,c-1}], \\ &\ldots, \\ &[\tilde{m}_{\ell-1,0},\, \tilde{m}_{\ell-1,1},\, \tilde{m}_{\ell-1,2},\, \ldots,\, \tilde{m}_{\ell-1,c-1}]]. \end{aligned}$$

(Note: essa forma de representar uma matriz privilegia as linhas em detrimento das colunas; poderíamos muito bem ter escolhido uma forma que privilegiasse as colunas em detrimento das linhas.)

a. Faça uma função que receba uma matriz na forma descrita acima e retorne a quantidade de linhas da matriz.

b. Faça uma função que receba uma matriz na forma descrita acima e retorne a quantidade de colunas da matriz.

c. Faça uma função que receba uma matriz na forma descrita acima e um número inteiro i e retorne a i-ésima linha da matriz dada, caso exista, ou algum tipo de erro informativo em caso contrário[1].

d. Faça uma função que receba uma matriz na forma descrita acima e um número inteiro j e retorne a j-ésima coluna da matriz dada, caso exista, ou algum tipo de erro informativo em caso contrário.

e. Faça uma função que receba uma matriz na forma descrita acima e retorne sua transposta.

f. No mesmo espírito, podemos representar um vetor real

$$v = (v_0, v_1, v_2, \ldots, v_{n-1})$$

em `python` usando a lista

$$[\tilde{v}_0,\, \tilde{v}_1,\, \tilde{v}_2,\, \ldots,\, \tilde{v}_{n-1}],$$

[1]No caso de erro, "retornado" deve ser entendido em sentido amplo, não necessariamente usando um `return`. O mesmo comentário vale para as questões seguintes.

onde novamente $\tilde{v}_i$ é o `float` que representa o real v_i.

Dados dois vetores reais de mesmo comprimento $u = (u_0, \ldots, u_{n-1})$ e $v = (v_0, \ldots, v_{n-1})$, seu *produto escalar* é o número real

$$\sum_{i=0}^{n-1} (u_i \cdot v_i).$$

Faça uma função que receba como entrada dois vetores (representados como acima) e retorne seu produto escalar, caso exista, ou algum tipo de erro informativo em caso contrário.

g. O *produto* de duas matrizes reais M e N existe quando (e apenas quando) o número de colunas de M é igual ao número de linhas de N. Neste caso, o produto é a matriz real cujo número de linhas coincide com o de M e cujo número de colunas coincide com o de N, e tal que o real que ocupa a i-ésima linha e j-ésima coluna é o produto escalar da i-ésima linha de M com a j-ésima coluna de N (vistos como vetores). Implemente essa operação em `python`, usando as representações descritas acima; caso o produto das matrizes dadas não exista, novamente, algum tipo de erro informativo deve ser retornado.

Parte 2 — Desafio opcional

Questão 4. Sobre a Questão b. Faça uma função que receba uma expressão bem formada `expr` e um natural `n` e retorne a menor subexpressão bem formada de `expr` que contenha (o símbolo na) a posição `n` da `expr` (começando a contagem de `0`, como sempre).

Por exemplo, com entradas `()(()(()()))` e `1`, a saída deve ser `()`, e com entradas `()(()(()()))` e `5`, a saída deve ser `(()())`.