

Ex2

Entrega 18 de ago de 2021 em 15:10 **Pontos** 2 **Perguntas** 1
Disponível até 18 de ago de 2021 em 15:20 **Limite de tempo** Nenhum
Tentativas permitidas Sem limite

Instruções

Para realizar esta tarefa, você:

- pode consultar a Bibliografia;
- não pode consultar colegas;
- deve observar o prazo máximo para responder, normalmente durante a aula.
- dê uma resposta objetiva, embora não haja limite mínimo ou máximo de conteúdo para sua resposta.

Este teste não está mais disponível, pois o curso foi concluído.

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	Tentativa 1	15 minutos	2 de 2

Pontuação desta tentativa: 2 de 2

Enviado 18 de ago de 2021 em 15:07

Esta tentativa levou 15 minutos.

Pergunta 1

2 / 2 pts

Considere os mecanismos de composição de tipos para as linguagens de programação. Escolha um, defina-o e mostre como o mesmo é implementado em um pequeno trecho de código de uma LP de sua preferência.

Sua resposta deve conter 4 componentes:

- 1) Mecanismo de composição escolhido;
- 2) Definição do mecanismo;
- 3) LP escolhida;
- 4) Trecho de código na LP escolhida.

Sua Resposta:

Uma matriz Python é uma matriz bidimensional de dados armazenados em linhas e colunas. Os dados em uma matriz podem ser números, strings, expressões, símbolos, etc. No exemplo escolhido, vamos fazer a multiplicação de matrizes utilizando recursão. Primeiro vamos verificar se a multiplicação entre matrizes é possível ou não. Para isso, verificamos se o número de colunas da primeira matriz é igual ao número de linhas da segunda matriz. Se ambos forem iguais, prossiga, caso contrário, gere uma saída dizendo ser impossível.

Na multiplicação de matriz recursiva, nós implementamos três loops por meio de chamadas recursivas.

A chamada mais recursiva interna de `multiplyMatrix()` é iterar `k` (`col1` ou `row2`).

A segunda chamada recursiva de `multiplyMatrix()` é para alterar as colunas e a chamada recursiva mais externa é para alterar as linhas.

```
def multiplyMatrix(row1, col1, A, row2, col2, B):  
    if (row2 != col1):  
        print("Not Possible")  
        return  
  
    C = [[0 for i in range(MAX)]  
          for i in range(MAX)]  
    multiplyMatrixRec(row1, col1, A,  
                      row2, col2, B, C)  
  
    # Print the result  
    for i in range(row1):  
        for j in range(col2):  
            print( C[i][j], end = " ")  
        print()
```

O mais importante aqui seria mostrar o carregamento das matrizes ou, neste caso, a matriz impressa também!

Pontuação do teste: **2** de 2