

2a. Prova de Programação de Computadores

Boa tarde, professora. Todas as questões foram respondidas com a função da biblioteca numpy, que normalmente é default do Python, ou seja, já vem instalada. Acredito que na sua IDL ela já esteja instalada também, mas, caso não, favor instalar **(FILE > PROJECT > PROJECT INTERPRETER > PIP > NUMPY)**, pois as funções rodam perfeitamente. Muito obrigado!!!

1a. Questão: Faça um algoritmo em Python que preencha uma matriz 4×4 com números inteiros aleatórios entre 1 e 10. Em seguida, crie um vetor que armazene os elementos da diagonal secundária dessa matriz. Imprima a matriz criada, o vetor e a soma dos 3 primeiros elementos desse vetor criado.

```
import numpy as np
```

```
matriz_aleatoria = np.random.randint(1,11,(4,4))
```

```
diag_sec = []
```

```
ix = -1
```

```
for i in matriz_aleatoria:
```

```
    elemento = i[ix]
```

```
    diag_sec.append(elemento)
```

```
    ix -= 1
```

```
print(f"A matriz criada é:\n\n{matriz_aleatoria}\n")
```

```
print(f"A diagonal secundária é {diag_sec}\n")
```

```
print(f"A soma dos três primeiros elementos ({diag_sec[0:-1]}) da diagonal é {sum(diag_sec[0:-1])}")
```

QUESTÃO 1

```
import numpy as np

matriz_aleatoria = np.random.randint(1,11,(4,4))

diag_sec = []
ix = -1
for i in matriz_aleatoria:
    elemento = i[ix]
    diag_sec.append(elemento)
    ix -= 1

print(f"A matriz criada é:\n\n{matriz_aleatoria}\n")
print(f"A diagonal secundária é {diag_sec}\n")
print(f"A soma dos três primeiros elementos ({diag_sec[0:-1]}) da diagonal é {sum(diag_sec[0:-1])}")
```

A matriz criada é:

```
[[ 8 10  8  8]
 [ 3 10  7  9]
 [10  2  6  1]
 [ 3  5  5  5]]
```

A diagonal secundária é [8, 7, 2, 3]

A soma dos três primeiros elementos ([8, 7, 2]) da diagonal é 17

2a. Questão: Faça um algoritmo em Python que preencha aleatoriamente um vetor de 10 posições com valores inteiros no intervalo [-5,5]. Crie uma função que recebe o vetor preenchido e substitua todas as ocorrências de valores menores que 0 por -1 e todos os valores maiores ou iguais a 5 por 1.

```
import numpy as np
```

```
matriz_aleatoria = np.random.randint(-5,6,(10))
```

```
def matriz_personalizada(matriz):
```

```
    menor_zero = np.where(matriz<0, -1, matriz)
```

```
    maior_igual_cinco = np.where(matriz>=5, 1, menor_zero)
```

```
    print(f"A matriz original é:\n\n{matriz}\n")
```

```
    print(f"Trocando valores menores que 0 por -1 e maiores ou iguais a 5 por 1...\n\n{maior_igual_cinco}")
```

matriz_personalizada(matriz_aleatoria)

QUESTÃO 2

```
import numpy as np

matriz_aleatoria = np.random.randint(-5,6,(10))

def matriz_personalizada(matriz):

    menor_zero = np.where(matriz<0, -1, matriz)
    maior_igual_cinco = np.where(matriz>=5, 1, menor_zero)

    print(f"A matriz original é:\n\n{matriz}\n")
    print(f"Trocando valores menores que 0 por -1 e maiores ou iguais a 5 por 1...\n\n{maior_igual_cinco}")

matriz_personalizada(matriz_aleatoria)
```

A matriz original é:

```
[-3 -5  2  3 -2 -4 -2  5  4  2]
```

Trocando valores menores que 0 por -1 e maiores ou iguais a 5 por 1...

```
[-1 -1  2  3 -1 -1 -1  1  4  2]
```

3a. Questão: Faça um programa em Python que preencha uma matriz 5×5 valores aleatórios entre 1 e 20, mostre qual é o maior valor existente na matriz desconsiderando os elementos da última linha.

import numpy as np

matriz_aleatoria = np.random.randint(1,21,(5,5))

valor_max = np.max(matriz_aleatoria[0:4])

print(f"A matriz é:\n\n{matriz_aleatoria}\n")

print(f"O máximo valor da matriz, desconsiderando a última linha é {valor_max}")

QUESTÃO 3

```
: import numpy as np

matriz_aleatoria = np.random.randint(1,21,(5,5))
valor_max = np.max(matriz_aleatoria[0:4])

print(f"A matriz é:\n\n{matriz_aleatoria}\n")
print(f"O máximo valor da matriz, desconsiderando a última linha é {valor_max}")
```

A matriz é:

```
[[13 12 16 17  6]
 [20  7  8 15 10]
 [19 14 14  9  4]
 [ 1 13  2 10  2]
 [12  1 11 14  3]]
```

O máximo valor da matriz, desconsiderando a última linha é 20

4a. Questão: Seja o seguinte programa abaixo, onde foi definida uma função **fat** que calcula o fatorial de um número inteiro não negativo. Execute o programa no pycharm e observe que ela não retorna o valor correto para o fatorial de um número. Encontre o erro na função. O comando de repetição utilizado dever ser o FOR (NÃO USE O WHILE).

```
def fat(x):
    if x<0:
        print('Fatorial não está definido para esse valor')
    if x<=1:
        return 1
    else:
        result=1
        for i in range(x):
            result=result*i
        return result
```

```
n=int(input('Digite um numero para o qual deseja determinar o fatorial'))
print(fat(n))
```

3 ERROS:

- 1) Valor inicial de “result” é o próprio parâmetro “result”, e não =1.
- 2) O range do loop “for” deve ser (1, x), e não (x), pois assim o número não é multiplicado por zero.
- 3) Erro de sintaxe: falta um parênteses no fechamento da declaração da variável “n”
`int(input(...)).`

FUNÇÃO CORRIGIDA:

```
def fat(x):  
    if x<0:  
        print('Fatorial não está definido para esse valor')  
    if x<=1:  
        return 1  
    else:  
        result=x  
        for i in range(1,x):  
            result=result*i  
        return result  
  
n=int(input('Digite um numero para o qual deseja determinar o fatorial'))  
print(fat(n))
```

QUESTÃO 4

```
def fat(x):  
    if x<0:  
        print('Fatorial não está definido para esse valor')  
    if x<=1:  
        return 1  
    else:  
        result=x  
        for i in range(1,x):  
            result=result*i  
        return result  
  
n=int(input('Digite um numero para o qual deseja determinar o fatorial'))  
print(fat(n))
```

Digite um numero para o qual deseja determinar o fatorial 2

2