

Lista 5: Matrizes

1. O que a função a seguir faz?

```
def DontPanic(S, n, m):
    R = [] * (n*m)
    for i in range(n):
        for j in range(m):
            R[i * m + j] = S[i][j]
    return R
```

2. Faça uma função que receba como parâmetros uma matriz quadrada e dois inteiros i e j . A função deve trocar os conteúdos das linhas i e j desta matriz entre si. Esta é uma operação de matrizes conhecida como permutação de linhas.
3. Faça uma função que receba como parâmetros uma matriz quadrada e dois inteiros i e j . A função deve encontrar o produto escalar entre as colunas i e j .
4. Faça um algoritmo que recebe uma matriz e devolve a matriz transposta.
5. Faça um algoritmo que recebe uma matriz e testa se ela é simétrica. Uma matriz é *simétrica* se ela é igual à sua transposta.
6. Faça um algoritmo que recebe uma matriz e verifica se existe um elemento dominante nela. Um elemento é *dominante* se todos os elementos em células adjacentes (até 8 elementos) são menores do que ele. Uma célula é *adjacente* a outra se estiver imediatamente à esquerda, à direita, acima, abaixo ou nas diagonais.
7. Faça um algoritmo que recebe uma matriz e verifica se ela é triangular superior. Uma matriz é *triangular superior* se só possui elementos nulos abaixo da diagonal principal.
8. Faça um algoritmo que recebe uma matriz e indica quantas colunas nulas e quantas linhas nulas ela possui.
9. Faça um algoritmo que verifica se há elementos repetidos em uma matriz. Caso haja, deve indicar em qual posição esses elementos estão.
10. Faça um algoritmo que recebe duas matrizes e verifica se a primeira está contida na segunda.
11. No Sudoku, uma matriz 9×9 é dividida em 9 submatrizes de tamanho 3×3 , chamadas de quadrantes. O objetivo do jogo é preencher os quadrados com números entre 1 e 9 de acordo com as seguintes regras: (i) cada número pode aparecer apenas uma vez em cada linha; (ii) cada número pode aparecer apenas uma vez em cada coluna; (iii) cada número pode aparecer apenas uma vez em cada quadrante. Faça um algoritmo que recebe uma matriz 9×9 e verifica se o seu conteúdo é um jogo válido de Sudoku.

12. O tabuleiro do jogo de *campo minado* consiste em uma matriz de N linhas e M colunas. Cada célula da matriz contém uma mina ou o número de minas que existem nas células adjacentes a ela. Uma célula é *adjacente* a outra se estiver imediatamente à esquerda, à direita, acima, abaixo ou nas diagonais. Note que, se não contiver uma mina, uma célula deve obrigatoriamente conter um número entre 0 e 8, inclusive.

Faça um algoritmo que, dadas as posições das minas, determine o tabuleiro do jogo.

A entrada portanto consiste inicialmente dos inteiros N e M ($1 \leq N, M \leq 1000$), e as próximas N linhas contêm M inteiros cada, descrevendo as minas de um campo: o j -ésimo inteiro da i -ésima linha é 1 se existe uma mina na linha i e coluna j do tabuleiro, e é 0 caso contrário.

Devem ser impressas N linhas com M inteiros cada, descrevendo a configuração do tabuleiro. Se uma posição contém uma mina, imprima * para ela; caso contrário, imprima o número cuja posição deve conter (algum número entre 0 e 8).

Veja um exemplo de entrada e saída abaixo.

ENTRADA	SAÍDA
4 4	12**
0 0 1 1	1*5*
0 1 0 1	34*3
0 0 1 0	**3*
1 1 0 1	

13. Há muito tempo atrás, em uma galáxia muito, muito distante...

Após o declínio do Império, sucateiros estão espalhados por todo o universo procurando por um sabre de luz perdido. Todos sabem que um sabre de luz emite um padrão de ondas específico: 42 cercado por 7 em toda a volta. Você tem um sensor de ondas que varre um terreno com $N \times M$ células. Veja o exemplo abaixo para um terreno 4×7 com um sabre de luz na posição (2, 4).

11	12	7	7	7	13	14
15	6	7	42	7	7	42
98	-5	7	7	7	42	7
-1	42	3	9	7	7	7

Você deve escrever um algoritmo que, dada a varredura de um terreno $N \times M$, procura pelo padrão do sabre de luz nele. Nenhuma varredura tem mais do que um padrão de sabre de luz.

A entrada portanto consiste inicialmente de dois números positivos N e M , representando, respectivamente, o número de linhas e de colunas varridos no terreno ($3 \leq N, M \leq 1000$). Cada uma das próximas N linhas tem M inteiros, que descrevem os valores lidos em cada célula do terreno ($-100 \leq T_{ij} \leq 100$, para $1 \leq i \leq N$ e $1 \leq j \leq M$).

A resposta consiste de 2 inteiros X e Y , com $1 \leq X \leq N$ e $1 \leq Y \leq M$. Eles representam a coordenada (X, Y) do sabre de luz, caso encontrado. Se o terreno não tem um padrão de sabre de luz, escreva Sabre não encontrado.

Veja exemplos a seguir.

ENTRADA	SAÍDA
4 7 11 12 7 7 7 13 14 15 6 7 42 7 7 42 98 -5 7 7 7 42 7 -1 42 3 9 7 7 7	2 4
4 7 11 12 7 7 7 13 14 15 6 7 12 7 7 42 98 -5 7 7 7 42 7 -1 42 3 9 7 7 7	Sabre nao encontrado
3 3 7 7 7 7 42 7 7 7 7	2 2

14. Historicamente César foi o primeiro a codificar mensagens. Ele reorganizava o texto de suas mensagens de maneira que o texto parecia não ter sentido. Cada mensagem sempre possuía uma contagem de letras cujo total equivalia a um quadrado perfeito, dependendo de quanto César tivesse que escrever. Assim, uma mensagem com 16 caracteres usava um quadrado de quatro por quatro; se fossem 25 caracteres, seria cinco por cinco; 100 caracteres requeriam um quadrado de dez por dez, etc. Seus oficiais sabiam que deviam transcrever o texto preenchendo as casas do quadrado sempre que uma mensagem cifrada chegasse. Ao fazerem isso, podiam ler a mensagem na vertical e seu sentido se tornaria claro.

Escreva um algoritmo que lê uma string e escreve a mensagem decifrada.

Por exemplo, se a string lida for “UEEUMOFSHMSCAT*AGUBA***LC****T*****A”, então ela pode ser transcrita em um quadrado 6×6 :

U	E	E	U	M	O
F	S	H	M	S	C
A	T	*	A	G	U
B	A	*	*	*	L
C	*	*	*	*	T
*	*	*	*	*	A