

Objectifs : savoir définir une tranche et une matrice

1. Soit la chaîne de caractères `date="2023-11-19"`. En utilisant des tranches, définir les entiers `a`, `m` et `j` correspondant à l'année, le mois et le jour.

Solution : `a, m, j = int(date[:4]), int(date[5:7]), int(date[8:])`

2. Ecrire une fonction `one(n)` retournant une matrice identité $n \times n$.

Solution : Par modification d'une matrice nulle :

```
1 def one(n):
2     M = [[0]*n for i in range(n)]
3     for i in range(n):
4         M[i][i] = 1
5     return M
```

ou par création directe (peut-être moins adapté ici) :

```
1 def one(n):
2     M = []
3     for i in range(n):
4         L = []
5         for j in range(n):
6             if i==j:
7                 L.append(1)
8             else:
9                 L.append(0)
10        M.append(L)
11    return M
```

3. Ecrire une fonction `degrade(n)` retournant une matrice carré $n \times n$ tel que $M_{i,j} = (i + 1) * (j + 1)$. On doit avoir, en particulier, $M_{0,0} = 1$.

Solution :

```
1 def degrade(n):
2     M = []
3     for i in range(n):
4         L = []
5         for j in range(n):
6             L.append((i+1)*(j+1))
7         M.append(L)
8     return M
```

4. Ecrire une fonction `taille(M)` retournant un couple (n,p) correspondant aux dimensions de la matrice définie par la liste de listes `M`.

Solution :

```
1 def taille(M):
2     return len(M), len(M[0])
```

5. Définir une fonction `colonne(M, j)` retournant la jème colonne (indicée à partir de 0) de la matrice M.

Solution : Colonne créée élément par élément ou par compréhension :

```
1 def colonne(M, j):
2     C = []
3     n = len(M)
4     for i in range(n):
5         C.append(M[i][j])
6     return C
7
8 def colonne(M, j):
9     n = len(M)
10    return [M[i][j] for i in range(n)]
```