

Requêtes SQL

1. Les espaces, l'indentation, les retours à la ligne, le choix de majuscule/minuscule n'importent pas.

a) `SELECT Title FROM publication;`

b)

```
SELECT Title, Year
FROM publication
ORDER BY Year ASC
LIMIT 5;
```

c)

```
SELECT Count(*)
FROM publication
WHERE Year > 1980;
```

d)

```
SELECT p.Title
FROM publication as p JOIN author as a ON a.Article = p.DOI
WHERE a.Name = "Curie" AND a.FirstName = "Marie";
```

e)

```
SELECT a.Name, p.Title
FROM publication as p JOIN author as a ON a.Article = p.DOI
WHERE a.Rank = 1 AND p.Year > 2000 AND p.Field = "Biology";
```

f)

```
SELECT a1.Name
FROM author as a1 JOIN author as a2 ON a1.DOI = a2.DOI
WHERE (a1.Name != "Turing" OR a1.FirstName != "Alan")
AND a2.Name = "Turing" AND a2.FirstName = "Alan";
```

Manipulation d'images

2.

a) M^{\leftrightarrow} a la même dimension que M , donc $m \times n$

b) $M_{i,j}^{\leftrightarrow} = M_{i,n-j}$

c)

```
def reflexion_horizontale(img):
    h, w = np.shape(img)
    ret = np.zeros((h, w))
    for i in range(h):
        for j in range(w):
            ret[i, j] = img[i, w-1-j]
    return ret
```

ou encore

```
def reflexion_horizontale(img):
    return img[:, ::-1]
```

3.

a) M^T a dimension $n \times m$.

b) $M_{i,j}^t = M_{j,i}$

c)

```
def transpose(img):
    h, w = np.shape(img)
    ret = np.zeros((w, h))
    for i in range(w):
        for j in range(h):
            ret[i, j] = img[j, i]
return ret

# ou encore
def transpose(img):
    return img.T
```

4.

a) M^\dagger a la même dimension que M , donc $m \times n$ b) $M_{i,j}^\dagger = M_{m-i,j}$

c) On montre l'égalité coefficient par coefficient :

$$\begin{aligned} (M^T)_{i,j}^{\leftrightarrow} &= M_{i,m-j}^T \quad \text{car } M^T \text{ est de dimension } n \times m \\ &= M_{m-j,i} \\ (M^\dagger)_{i,j}^T &= M_{j,i}^\dagger \\ &= M_{m-j,i} \end{aligned}$$

d) Il suffit de prendre la transposée de l'égalité prouvée à la question précédente.

5.

```
def reflexion_verticale(img):
    return transpose(reflexion_horizontale(transpose(img)))

# pas la solution attendue, mais on peut aussi écrire
def reflexion_verticale(img):
    return img[::-1, :]
```