DS09

## Question 1

```
SELECT idpatient
FROM MEDICAL
WHERE etat = "hernie discale"
```

✓

## Question 2

Il manque la jointure

```
SELECT nom, prenom
FROM PATIENT, MEDICAL
WHERE etat = "spondylolisthésis"
```

## Question 3

```
SELECT etat, COUNT(*) AS nb_etat
FROM MEDICAL
GROUP BY etat
```

Il manque le distinct

## Question 4

Lorsque les tableaux sont de grande taille les calculs ou algorithmes peuvent être long et peu efficace, donc la bibliothèque Wumpy peut venir en aide avec les fonctions qu'elle contient.

✓

## Question 5

On a 100 000 lignes et 6 colonnes.
Il y a donc 600 000 cases.

Le tableau est codé avec 32 bits donc chaque case prend 4 octets de mémoires.
Chaque valeur dans chaque case prend 8 ~~octets~~ bits et il y en a N à stocker.

Donc $4 \times 600\,000 + 100\,000 = 2,5$ Mo

Le tableau va prendre 2,5 Mo

V

## Questions 6

```
def separationParGroupe(data, etat):
    liste = [[], [], []]
    for i in range(len(etat)):
        if etat[i][0] == 0:
            liste[0].append(data[i])
        elif etat[i][0] == 1:
            liste[1].append(data[i])
        elif etat[i][0] == 2:
            liste[2].append(data[i])
    return liste
```

## Question 12

La partie 1 de la fonction ~~met~~ met dans la ~~tab~~ liste T les distances euclidiennes (dist[i]) avec le numéro de la distance. Elle classe ou range chaque distance dans l'ordre croissant ~~avec le format [ , i ]~~

Partie 2 : ~~distance~~ met dans la table ~~selection~~ select le nombre des K plus proches, ~~en fonction de leur positions~~ en fonction de leurs positions dans la table T.

Partie 3 : Prendre la plus grande valeur de la liste sect donc là où il y a le plus d'entiers proches.

## Question 14

On obtient une courbe qui nous montre que le ~~taux~~ taux de reussite ~~est~~ est entre 68 et 74 %
Donc l'algorith est plutot efficace

V

## Question 15

```
def moyenne (x):
    return sum(x)/len(x)
```

Interdit

```
def variance (x):
    T = []
    for i in len(x):
        t = (x[i] - mean(x)) ** 2
        T.append (t)
    return sum (T) /(len(x)-1)
```