

BERNARD
Thiophane

MPSI₂

DS n°9 d'informatique.

Q1- SELECT idpatient FROM MEDICAL
WHERE etat = "hernie discale"; ✓

Q2- SELECT PATIENT.nom, PATIENT.prenom
FROM PATIENT
JOIN MEDICAL ON PATIENT.id = MEDICAL.id-
patient WHERE etat = "mandibulosthéus"; ✓

Q3- SELECT etat, count(*) as nbpatient
FROM MEDICAL Distinct
JOIN PATIENT ON MEDICAL.idpatient = PATIENT.id
GROUP by etat;

Q4- L'intérêt de mumpy quand le tableau
est de grande taille est que l'on peut accéder
simplement aux différentes valeurs de celui-ci.

Pourquoi ?

Question 5-

Quantité de bits pour tout stocker :

$$\begin{aligned} & 6 \times 100\,000 \times 32 + 100\,000 \times 8 \\ &= 3\,200\,000 \times 6 + 800\,000 \\ &= 19\,200\,000 + 800\,000 \\ &= 20\,000\,000 \text{ bits} \\ &= 2\,500\,000 \text{ octets} \\ &= 2,5 \text{ Mo.} // \end{aligned}$$



Q6-

def separationParGroupe(data, etat):

data0 = []

data1 = []

data2 = []

for k in range(N):

if etat[k] == 0:

data0.append(data[k])

elif etat[k] == 1:

data1.append(data[k])

else:

data2.append(data[k])

return [data0, data1, data2]

Attention à
l'indentation

Q7- ARG1: $n, n, (i \times n) + j + 1$

TEST: $i \neq j$

ARG2: $\text{groupes}[k][:, i], \text{groupes}[k][:, j],$
 $\text{marker} = \text{mark}[k]$

ARG3: $\text{groupes}[0][:, i] + \text{groupes}[1][:, i]$
 $+ \text{groupes}[2][:, i]$

En python ?



Q8- Les diagrammes de la diagonale servent à se rendre compte du ~~nombre~~ nombre de patients ayant une caractéristique pour un attribut donné. ✓

Les diagrammes hors diagonale représentent comment se répartissent les patients en fonction de deux caractères donnés et en fonction de leur état.

Q9- x_{ij} et $x_{norm j} = \frac{x_j - \min(X)}{\max(X) - \min(X)}$ ✓

Q10-

```
def min_max(X):  
    min = X[0]  
    max = X[0]  
    for k in range(1, N):  
        if X(k) > max:  
            max = X(k)  
        elif X(k) < min:  
            min = X(k)  
        else:  
            pass  
    return min, max
```

Attention à l'indentation

Complexité ?

Q11-

```
def distance(z, data):  
    L = []  
    for k in range(N):  
        a = 0  
        for i in range(N):  
            a = a + abs(z[i] - data[k][i])
```

Il manque le carré

L.append(a)
return L.

Q12 - Partie 1:

Cette partie permet de créer la liste T à 2 éléments, que l'on doit trier, présentée dans l'énoncé.

T représente la distance entre la 1^{re} ligne de état et z ainsi que l'indice de cette ligne.

$dist$ représente la liste de toutes les distances entre les patients et le cas à cheval.

Partie 2:

Cette partie permet de dénombrer le nombre de patients par ~~variable~~ état.

$select$ est la liste représentant ces nombres de patients par état.

Partie 3:

permet de déterminer quel est l'état qui a le plus de patients.

La variable ind est le ce numéro d'état.

Q13 - La diagonale de la matrice nous renseigne le ~~nombre~~ nombre d'états qui ont été correctement prédits.

23 patients avaient comme état 1 et ont été prédit 1

4 patients avaient comme état 1 et ont été prédit 2

7 patients avaient comme état 1 et ont été prédit 3.

7 patients avaient comme état 2 et ont été prédit 1.

5 patients avaient comme état 3 et ont été prédit 1.

Donc cette matrice nous renseigne comment ont été prédit les états des patients en fonction de leur vrai état.