



**Concours Biologie Géologie**  
**Alternative de correction de l'épreuve d'Informatique**

**Barème sur 100**

**PROBLEME 1 (100 pts)**

**Question 1 : 5 pts**

```
import sqlite3
base=sqlite3.connect('hopital.db')
curseur = base.cursor()
curseur.execute(' ' ' CREATE TABLE IF NOT EXIST Plat
(IdPlat INTEGER PRIMARY KEY , Nom TEXT , Calories
INTEGER; ' ' ' )
curseur.commit()
curseur.close()
base.close()
```

**Question 2 : 2,5 pts**

```
INSERT INTO Patient VALUES (1256 , 'BEN FOULEN' , 'Mohamed'
, 61 , 'Tunis' , '26/05/2017' , NULL) ;
```

**Question 3 : 2,5 pts**

```
UPDATE Plat SET Calories = 1200
WHERE IdPlat = 26 ;
```

**Question 4 : 2,5 pts**

```
DELETE FROM Plat

WHERE Calories < 400 ;
```

**Question 5 : 2,5 pts**

```
SELECT *

FROM Patient ;
```

**Question 6 : 2,5 pts**

```
SELECT IdPatient , Nom , Prenom , Date_Entree  
FROM Patient  
WHERE Age BETWEEN 25 AND 50 ;
```

**Question 7 : 5 pts**

```
SELECT SUM(Calories)  
FROM Servi , Patient , Plat  
WHERE Servi.IdPlat = Patient.Idplat AND Nom = 'FALTENA' AND  
Prenom = 'Fattouma' AND Plat.IdPlat = Servi.IdPlat ;
```

**Question 8 : 5 pts**

```
SELECT IdPatient , SUM(Calories)  
FROM Servi , Plat  
WHERE Servi.IdPlat = Plat.Idplat AND Date = '26/05/2017' AND  
Quand = 'P'  
GROUP BY IdPatient  
ORDER BY SUM(Calories) DESC;
```

**Question 9 : 2,5 pts**

```
SELECT Nom , Calories  
FROM Plat  
WHERE Calories = (SELECT MAX(Calorie) FROM Plat);
```

**Question 10 : 5 pts**

```
SELECT Nom , Prenom  
FROM Servi , Patient , Plat  
WHERE Servi.IdPatient = Patient.IdPatient AND Quand = 'S' AND  
Date = '25/05/2017' AND Servi.IdPlat = Plat.IdPlat  
AND Plat.Nom = 'Soupe' ;
```

**Question 11 : 2,5**

```
SELECT Nom , Prenom , IdPatient  
FROM Patien  
WHERE Date_Sortie IS NULL ;
```

**Question 12 : 2,5 pts**

```
SELECT Nom
FROM Plat
WHERE IdPlat NOT IN (SELECT DISTINCT (IdPlat) FROM SERVI ;
```

**Question 13 : 5 pts**

```
SELECT Patient.Nom , Prenom , Patient.IdPatient
FROM Interdit , Patient , Plat
WHERE Interdit.IdPlat = Plat.IdPlat AND Interdit.IdPatient =
      Patientt.IdPatient AND Plat.Nom = 'Salade
      Tunisienne' ;
```

**Question 14 : 5 pts**

```
SELECT COUNT(*)
FROM Patient
WHERE IdPatient NOT IN (SELECT DISTINCT (IdPatient) FROM
      Interdit ;
```

**PROBLEME 2 (100 pts)**

**Question 1 : 10 pts**

```
def    saisiEquipes(N):
    L = []
    for i in range(N):
        while True:
            E=input(" donner une équipe : ")
            if E not in L:
                L.append(E)
                break
    return L
```

**Question 2 : 5 pts**

```
def    genRencontre(LE):
    LR = []
    for i in range(len(LE) - 1):
        for j in range(i + 1 , len(LE)):
            LR.append((i , j))
    return LR
```

**Question 3 : 10 pts**

```
def    saisiResultats(LE):
    LRES = genRencontre(LE)
    for    i    in    range(len(LRES)):
        while True:
            while (True):
                try:
                    a = int(input("résultat de la {}
ème rencontre: ".format(i+1)))
                    if    a    in    (0 , 1 , 2):
                        break
                except:
                    print('erreur de saisie du
résultat')
            while True:
                try:
                    b,c = eval(input("buts  de la
rencontre {}: ".format(i+1)))
                    if type(b)==int and type(c)==int:
                        break
                except:
                    print('erreur de saisie des buts')
                    if (a == 0 and b==c) or (a == 1 and b>c) or
(a == 2 and b<c):
                        LRES[i]+=a,b,c
                        break
            else:
                print('erreur incoherence')
    return LRES
```



**Question 4 : 10 pts**

```
def resultatsEquipe(LRES, LE) :
    LTR = []
    for i in range(len(LE)) :
        a,b,c,d,e,f=len(LE)-1,0,0,0,0,0
        for t in LRES:
            if t[0]==i:
                if t[2]==1:
                    b+=1
                elif t[2]==0:
                    c+=1
                else:
                    d+=1
                e+=t[3]
                f+=t[4]
            if t[1]==i:
                if t[2]==2:
                    b+=1
                elif t[2]==0:
                    c+=1
                else:
                    d+=1
                e+=t[4]
                f+=t[3]
        LTR.append((a,b,c,d,e,f))
    return LTR
```

**Question 5 : 5 pts**

```
def resultats(LTR) :
    LP=[]
    for i in range(len(LTR)) :
        p = LTR[i][1] * 3 + LTR[i][2]
        LP.append(p)
    return LP
```

**Question 6 : 10 pts**

```
def resultats(LP , LE) :
    L = []
    for i in range(len(LP)) :
        L.append((LP[i] , LE[i]))
    L=sorted(L , reverse = True)
    for p , e in L :
        print(e, "\t",p,"\n")
```