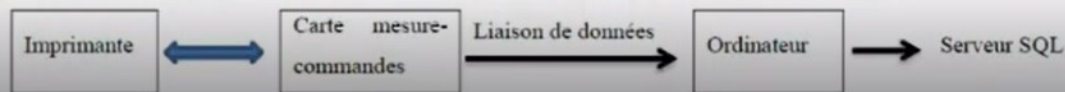


## Exercice 02

Les imprimantes sont des systèmes mécatroniques fabriqués en grande série dans des usines robotisées. Pour améliorer la qualité des produits vendus, il a été mis en place différents tests de fin de chaîne pour valider l'assemblage des produits. Pour un de ces tests, un opérateur connecte l'outil de test sur la commande du moteur de déplacement de la tête d'impression et sur la commande du moteur d'avance papier. Une autre connexion permet de récupérer les signaux issus des capteurs de position.

Différentes commandes et mesures sont alors exécutées. Ces mesures sont envoyées par liaison de données sous la forme d'une suite de caractères ASCII (chaîne de caractères) vers un ordinateur.

Cet ordinateur va effectuer différentes mesures pour valider le fonctionnement de l'électromécanique de l'imprimante. L'ensemble des mesures et des analyses est sauvegardé dans un fichier texte.



La commande `carac_recus=com.read(nbre_car)` permet de récupérer `nbre_car` caractères reçus sous la forme d'une chaîne de caractères. En supposant que les caractères reçus correspondent à l'exemple précédent, après l'exécution de `carac_recus=com.read(5)`, la variable `carac_recus` contiendra la chaîne "U005+". Après une nouvelle exécution de `carac_recus=com.read(3)`, la variable `carac_recus` contiendra la chaîne "012".

**Q1.** Ecrire une fonction `lect_mesures()` en langage Python qui retourne une liste contenant : le type de la mesure ('U', 'I' ou 'P'), une liste contenant l'ensemble des valeurs des mesures reçues et le checksum. Exemple : ['U',[12.4,-23,-2.42],83]. Cette fonction doit attendre que le caractère d'entête reçu soit correct ('U', 'I' ou 'P') avant de réaliser le stockage des informations dans la liste qui sera retournée.

**Q2.** On suppose que toutes les mesures sont disponibles dans la liste `mesure`, et le checksum reçu dans la variable `Checksum`. Ecrire une fonction `check(mesure,Checksum)` en langage Python qui retourne `True` si la transmission présente un checksum valide et `False` sinon.

**Q3.** Ecrire une requête SQL permettant d'obtenir les numéros de série des imprimantes ayant une valeur de `Imoy` comprise strictement entre deux bornes `Imin` et `Imax`.

**Q4.** Sachant que le chemin de cette base de données dans le disque dur est "C ://TestsImpr.sqlite", et en utilisant le module `sqlite3` de python, écrire un *programme python* permettant de se connecter à cette base de données, d'exécuter la requête *de la question précédente*, de récupérer le résultat et de :

- L'afficher sur l'écran.
- Le stocker dans un fichier.

#Q1

```
def lect_mesures():
    L=[]
    carac_recus=com.read(1)
    while carac_recus!=" " and carac_recus not in ('U','I','P'):
        carac_recus=com.read(1)
    if carac_recus in ('U','I','P'):
        L.append(carac_recus)
        Lm=[]
        nbr_m=int(com.read(3))
        for i in range(nbr_m):
            Lm.append(int(com.read(4)))
        L.append(Lm)
        L.append(int(com.read(4)))
    return L
```

#Q2

```
def check(mesure,Checksum):
    s=0
    for m in mesure:
        s+=abs(m)
    return s%10000==Checksum
```

#Q3

```
req="select nSerie from testfin where Imoy>=Imin and Imoy<=Imax"
```

#Q4

```
c=connect("C://TestsImpr.sqlite")
cur=c.cursor()
res=cur.execute(req)
nomf=input("entrer le nom:")
f=open(nomf,"w")
for ligne in res:
    print(ligne)
    f.write(str(ligne)+"\n")
f.close()
```

**Exercice 03(Médian d'une liste de nombres) :**

$L$  est une liste de taille  $n$  qui contient des nombres, et  $m$  un élément de  $L$ .

L'élément  $m$  est un **médian** de  $L$ , si les deux conditions suivantes sont vérifiées :

- Le nombre d'éléments de  $L$ , qui sont inférieurs strictement à  $m$ , est inférieur ou égale à  $n/2$
- Le nombre d'éléments de  $L$ , qui sont supérieurs strictement à  $m$ , est inférieur ou égale à  $n/2$

**Exemple:** On considère la liste  $L = [ 25 , 12 , 6 , 17, 3 , 10 , 20 , 12 , 15 , 38 ]$ , de taille  $n=10$ . L'élément 12 est un médian de  $L$ , car :

- 3 éléments de  $L$  sont inférieurs strictement à 12, et  $3 \leq n/2$  ;
- 5 éléments de  $L$  sont supérieurs strictement à 12, et  $5 \leq n/2$ .

Q5. Écrire la fonction **median(L)** qui reçoit en paramètre une liste de nombres  $L$  non vide, et qui renvoie un élément médian de la liste  $L$ .

Q5

```
#exercice05
#Q1
def median(L):
    n=len(L)
    for m in L:
        a=grands(m,L)
        b=petits(m,L)
        if a<=n//2 and b<=n/2:
            return m
def grands(m,L):
    c=0
    for x in L:
        if m<x:
            c+=1
    return c
def petits(m,L):
    return len(L)-grands(m,L)-L.count(m)
```