TD11: Prog.

3.1 Fusion

On considère des fichiers tels que :

- chaque ligne contient un entier (pour simplifier)
- le fichier est trié par valeur croissante des entiers

On souhaite fusionner 2 tels fichiers, c'est à dire écrire un 3ème fichier qui réunit les entiers contenus dans les 2 fichiers, dans l'ordre croissant. On ne se préoccupera pas des éventuels doublons. Par exemple, si le 1er fichier contient les entiers `1 4 6 12` et le 2ème fichier contient les entiers `1 2 3 6 20 22` on trouvera dans le 3ème fichier les entiers `1 2 3 4 6 6 12 20 22`.

4. Proposer une solution qui lit le contenu des fichiers puis passe par un calcul sur une structure de données intermédiaire avant d'écrire le résultat dans un 3ème fichier.
5. Proposer une solution qui ne passe pas par une structure de données intermédiaire.

4.
```
def fusion(df1: TextIOWrapper, df2 : TextIOWrapper, sortie_)-> None.
    contenu_f1 = df1.readlines()
    contenu_f2 = df2.readlines()
    while contenu_f1 != [] and contenu_f2 != []:
        if contenu_f1[0] < contenu_f2[0]:
            sortie.write(contenu_f1.pop(0))
        else:
            sortie.write(contenu_f2.pop(0))
    while contenu_f1 != []:
        sortie.write(contenu_f1.pop(0))
    while contenu_f2 != []:
        sortie.write(contenu_f2.pop(0))
```

5. def fusion(df1: TextIOWrapper, df2: TextIOWrapper, sortie_) -> None.

```
    contenu_f1 = df1.Readline()
    contenu_f2 = df2.Readline()
    Res = ""
    while contenu_f1 != "\n" and contenu_f2 != "\n"
        if contenu_f1 < contenu_f2
            Res += contenu_f1
            contenu_f1 = df1.Readline()
        else:
            Res+= contenu_f2
            contenu_f2 = df2.Readline()
    while contenu_f1 != "\n":
        Res += contenu_f1
        contenu_f1 = df1.Readline()
    while contenu_f2 != "\n":
        Res+= contenu_f2
        contenu_f2 = df2.Readline()
    sortie.write(Res)
```

3.2 Précipitations

On considère un fichier de données `precipitations.csv` qui contient des dates, des mesures de précipitations (on mesure la pluie qui est tombée en millimètres) et de mesure de vent (en km par heure). Par exemple :

```
date;precipitations;vent
20231120;1.4;45.0
20231121;1.4;60.1
20231121;1.4;5.2
20231122;1.2;10.
20231123;0;2.
```

6. Proposer une solution qui permet :

- de calculer la moyenne des précipitations contenues dans le fichier
- de vérifier si oui ou non les dates sont triées par ordre croissant
- de vérifier si oui ou non les dates sont toutes différentes
- d'ajouter une ligne contenant une date et deux mesures (précipitations et vent) au fichier

6.a
```
def moyenne(df: TextIOWrapper) -> float:
    data = df.readlines()[1:]  # len(data) > 0.
    moyenne = 0.
    for line in data.split(";"):
        moyenne += float(line[2])
    Return moyenne / len(data).
```

b.
```
def dates_triees(df: TextIOWrapper) -> bool:
    data = df.readlines()[1:]  # len(data) > 0.
    for i in range(len(data)-1):
        line = data[i].split(";")
        following_line = data[i+1].split(";")
        if line[0] > following_line[0]:
            Return False
    Return True.
```