Exercice sur les listes et les chaines de caractères

a) Soit une chaine *ch* contenant des informations sur une personne : son nom, son prénom, son département de naissance et son année de naissance (par exemple : "Dupuis Alain 56 1998"). On suppose que *ch* contient exactement 4 valeurs séparées les unes des autres par un espace.
Ecrire un programme qui affiche **l'année de naissance** (c'est-à-dire les 4 derniers caractères de *ch*).

```
print(ch[-4:])

# ou bien

print(ch[-4]+ch[-3]+ch[-2]+ch[-1])
```

b) Soit la même chaine *ch* que précédemment. Écrire un programme qui **affiche le prénom** de la personne (sans espace entre les lettres). Ecrire **une version** dans laquelle on parcourt la chaine *ch* (avec une boucle *while* et une variable *i*) de manière à extraire les caractères constituant le prénom et une **autre version** qui utilise la méthode *.split().*

```
print(ch.split()[1])

#ou bien

i = 0
while ch[i] != " " :
    i = i + 1
resu = ""
i = i + 1   # attention à ne pas oublier cette incrémentation …
while ch[i] != " " :
    resu = resu + ch[i]
    i = i + 1
print(resu)
```

c) Soit la même chaine *ch* que précédemment. Soit la liste *dpt* qui contient : ['Morbihan', 56, 'Finistère', 29, "Côtes-d'Armor", 22, 'Ille-et-Vilaine', 35]. Ecrire un programme **qui affiche le nom du département** correspondant au numéro du département de naissance de la personne. **Attention**, les numéros de département dans la liste *dept* sont des entiers et non des chaines comme dans *ch*. Sur l'exemple précédent, on obtient : Morbihan.

```
dep = ch[-7:-5]
i = 1
while i < len(dpt):
    if int(dep) == dpt[i]:    # attention à ne pas oublier le int()
        print(dpt[i-1])
    i = i + 1
```

d) Soit une liste *listeP* contenant **des** chaines de caractères composées chacune de quatre valeurs : un nom, un prénom, un département naissance et une année naissance). Ecrire un programme qui affiche **l'année de naissance la plus « ancienne » "** ainsi que **l'âge** des personnes nées cette année-là. **Utiliser une boucle *for element in listeP*** (et non une boucle qui gère les positions des éléments dans la liste).

Exemple :
Avec : listeP = ["Dupuis Alain 56 1995", "Durant Killian 35 2000", "Dupont Guillian 56 1995", "Dubois Marie 56 1997"]
On doit afficher :
1995
26 ans

```
min = 2021
for e in listeP:
    an = e[-4:]
    if int(an) < min :
        min = int(an)
age = 2021 - min
print(min)
print(age, 'ans')
```

e) Compléter le programme précédent de manière à afficher les **noms et prénoms des personnes les plus âgées**. **Solution simple** : ajouter une deuxième boucle **à la suite** de la boucle déjà écrite (autrement dit, avoir 2 boucles qui se suivent, et pas 2 boucles imbriquées). **Solution plus compliquée** (pour les étudiants qui seraient en avance) : tout faire dans une seule et même boucle.
Avec la liste donnée en exemple plus haut, on obtient :
Dupuis Alain
Dupont Guillian

On peut ajouter ce qui suit à la suite du programme précédent :

```
for e in listeP:
    an = e[-4:]
    if int(an) == max:
        p = e.split()
        print(p[0], p[1])

# ou alors (mais plus compliqué) :

min = 2021
personnes = ""
for e in listeP:
    an = e[-4:]
    if int(an) < min :
        min = int(an)
        personne = e.split()[0] + " " + e.split()[1]
    else:
        if int(an) == min :
            personne = personne + "\n" + e.split()[0] + " " + e.split()[1]
age = 2021 - min
print(min)
print(age, 'ans')
print(personne)
```

f) Soit la liste *listeP* de la question précédente. Construire une nouvelle liste dans laquelle tous les **noms de famille sont écrits en majuscules**. Sur l'exemple précédent, on obtient :
['DUPUIS Alain 56 1995', 'DURANT Killian 35 2000', 'DUPONT Guillian 56 1995', 'DUBOIS Marie 56 1997']

```
newListe =[]
for e in listeP:
    listeElem = e.split()
    nomEnMajus = listeElem[0].upper()
    newListe = newListe + [nomEnMajus + " "+ listeElem[1] + " " + listeElem[2] + " "
+ listeElem[3]]
print (newListe)
```