

**Exercice 1 :**

- 1)
- 2)
- 3)
- 4) La chargeFichierCsv s'écrit et se teste avec :

```
fonctions_fichiers_csv.py
1 import csv
2
3 def chargeFichierCsv(nom : str) -> list[list[str]] :
4     # Liste pour stocker les lignes
5     lignes: list[str] = []
6     # ouverture
7     fichier = open(nom, "r", encoding="utf-8")
8     reader = csv.reader(fichier,delimiter=';') # lecture
9     for ligne in reader:
10         lignes.append(ligne) # Ajouter chaque ligne à la liste
11     fichier.close() #fermeture.
12     return lignes
13
14 if __name__=="__main__" :
15     data=chargeFichierCsv("nat2021.csv")
16     assert (data[500]==['1', 'ABD', '1954', '20'])
17     assert(len(data)==686539)
18     print(len(data))
19     print(data[500])
20     # AJOUTER VOS autres tests ici
```

Avant de passer à la suite il est indispensable que vous soyez convaincus de la validité et conformité des données que vous chargez.

5) Votre fonction de la question 4 va être importée dans un autre code. Vous allez importer le module écrit précédemment. Vous allez tester que l'importation se passe bien.

EX\_1\_prenoms.py

```
1
2 import fonctions_fichiers_csv as fic
3
4 def testImport(nom_fichier : str)->list[list[str]] :
5     """Cette fonction permet de vérifier l'import
6     correct du module et de son bon fonctionnement"""
7     data : list[str] =fic.chargeFichierCsv(nom_fichier)
8     print(f" len(data) = {len(data)}")
9     return data
10
11 nom_fichier="nat2021.csv"
12 data : list[list[str]] = testImport(nom_fichier)
13 assert (data[500]==['1', 'ABD', '1954', '20'])
14 assert(len(data)==686539)
15 print(len(data))
16 print(data[500])
17 # AJOUTER VOS TESTS ICI (vérifier une autre ligne de data par exemple)
```

6) Voici la fonction extrait data complète, toujours avec ses tests.

```
1 def extrait_data(data:list[list[str]]) :  
2     """Cette fonction extrait les données  
3     et renvoie 4 listes de même longueur"""  
4     sexe : list = []  
5     prenom :list = []  
6     annees :list = []  
7     nb :list = []  
8     for k in range(len(data)):  
9         ligne=data[k]  
10        sexe.append(ligne[0])  
11        prenom.append(ligne[1 ])  
12        annees.append(ligne[2])  
13        nb.append(ligne[3])  
14    return sexe, prenom, annees, nb  
15  
16 #nom_fichier="nat2021.csv"  
17 #data : list[list[str]] = testImport(nom_fichier)  
18 #assert (data[500]==['1', 'ABD', '1954', '20'])  
19 #assert(len(data)==686539)  
20 #print(len(data))  
21 #print(data[500])  
22  
23 nom_fichier="nat2021.csv"  
24 data : list[str] =fic.chargeFichierCsv(nom_fichier)  
25 sexe, prenom, annees, nb = extrait_data(data)  
26 assert(prenom[500] =='ABD')  
27 assert(annees [500]== '1954')  
28 assert(nb[500] = '20')
```

## 7) Recopier , exécuter, tester dans votre fichier E

```
code à rajouter
1 def extraitDonneesPourPrenom(PrenomCherche : str , Genre: int , sexe:list[str],
2     prenom :list[str], annees:list[str],nb:list[str]) :
3     """Affiche le nombre de fois où un prénom est donnée en fonction
4     de l'année de naissance
5     Retourne la liste des années et des effectifs correspondants
6     """
7     ans      : list[int]=[]      # années où le prénom a été donné
8     effectifs : list[int]=[]      # nombre du prénom cherché dans l'année.
9
10    for k in range(len(prenom)):
11        if str(prenom[k])== str(PrenomCherche) and Genre==int(sexe[k]) :
12            if annees[k]!='XXXX':
13                ans.append(int( annees[k] ))
14                effectifs.append(int( nb[k] ))
15
16    return ans,effectifs
17
18 PrenomCherche='CAMILLE' # prénom épïcène, prévoir les 2 cas de recherche.
19 Genre=2
20 ans2,effectifs2 = extraitDonneesPourPrenom(PrenomCherche, Genre,
21                                             sexe, prenom, annees, nb)
22 Genre=1
23 ans1,effectifs1 = extraitDonneesPourPrenom(PrenomCherche, Genre,
24                                             sexe, prenom, annees, nb)
```

A vous de tester dans le prompt les commandes suivantes :

```
1 >>> print(len(ans2) , len(effectifs2) )
2 >>>print(ans2)
```

Regarder ce que vous obtenez , ouvrez le fichier avec gedit ...bref tous les essais sont biens. Vous devez comprendre et être sûr(e)s de vos données. Ne rien tester est une très mauvaise pratique. Les bugs futurs seront encore plus chronophage si vous ne testez rien au fur et à mesure.

8) Un lot possible de commande matplotlib est le suivant. Par rapport au CM vous noterez que pour tracer 2 courbes, il y a 2 commandes plt.plot qui se suivent. N'hésitez pas à modifier les options, les couleurs etc ...

Vous noterez le passage des paramètres, exemple color s'écrit color='red'. Le nom du paramètre est précisé lors de l'appel. Nous en reparlerons en cours.

```
1 PrenomCherche='CAMILLE'
2 Genre=1 # H
3 ans1,effectifs1 = extraitDonneesPourPrenom(PrenomCherche, Genre, sexe, prenom,
4   annees, nb)
5 label1 = PrenomCherche + ' (H) '
6
7 PrenomCherche='CAMILLE'
8 Genre=2 # H
9 ans2,effectifs2 = extraitDonneesPourPrenom(PrenomCherche, Genre, sexe, prenom,
10  annees, nb)
11 label2 = PrenomCherche + ' (F) '
12
13 plt.plot(ans2,effectifs2, label=label2, color='red')
14 plt.plot(ans1,effectifs1, label=label1 ,color='blue')
15 plt.xlabel('Ans ')
16 plt.ylabel("Nombre du prénom cherché dans l'année.")
17 plt.title(f'Nombre du prénom {PrenomCherche} par an en France.')
18 plt.legend()
19 plt.grid()
20 plt.show()
```

9) A vous de jouer.

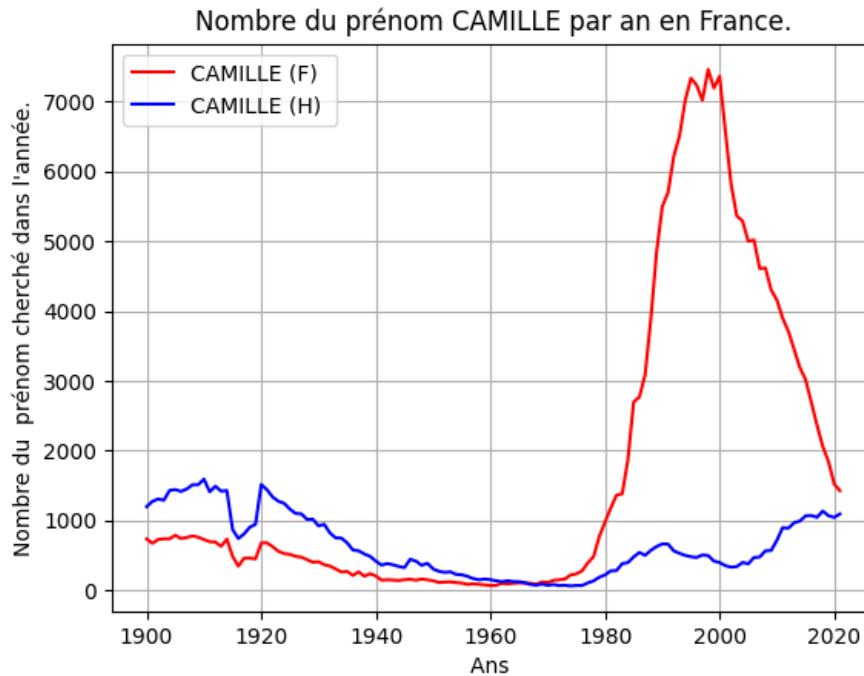


FIGURE 1 – Représentation graphique du nombre de prénom par an.

## Exercice 2 :

### Travail à faire :

- Créer un nouveau, fichier nommé EX2.py.
- Copier et coller dans EX2.py le code suivant.
- Ajouter vos lignes de code après chaque question.
- Faire pour chaque question des affichages explicites, en affichant le numéro de la question et les valeurs obtenues. Rédiger de façon à ce que l'on lise clairement ce que vous avez obtenu.

```
1 import numpy as np
2 # 1. Créer le tableau x=[ 1 2 3 4 5 ]
3 listeVal = [ 1, 2 ,3 ,4 ,5 ]
4 x=np.array(listeVal)
5 print(f"1) x = {x}") # affichage explicite du résultat.
6 print() # pour laisser une ligne vide.(+ lisible.)
7 # 2. Afficher le type de x puis sa longueur.
8
9 print(f"2) Le type de x est {type(x)}.") # affichage du résultat.
10 print()
11
12 # 3. Extraire le premier élément, puis en faire de même avec le dernier.
13 # Afficher les résultats dans une phrase explicite.
14 p=x[0]
15 f=x[-1]
16 print(f"3) Le premier élément x[0]= {p} et le dernier x[-1] ={f}.")
17 print()
18 # 4. Extraire les trois premiers éléments et les stocker dans un tableau
19 # que l'on nommera a.
20 # Afficher les résultats dans une phrase explicite.
21 a=x[0:3]
22 print(f"4) Les trois premiers éléments de x sont a=x[0:3] soit a ={a}.")
23 print()
24
25 # 5. Extraire les 1er, 2e et 5e éléments du tableau (attention aux positions)
26 # les stocker dans un tableau que l'on nommera b.
27 # Afficher les résultats dans une phrase explicite.
28 listeIndice=[0 ,1 ,4 ] # le premier est l'indice 0 !!
29 b=x [ listeIndice ]
30
31 print(f"5) Les 1er, 2e et 5e éléments de x sont b = x [ [0 ,1 ,4 ] ] soit b={b}. ")
32 print()
```

```
1 # 6. Additionner le nombre 10 au tableau x,  
2 # puis multiplier le résultat par 2.  
3 # Afficher les résultats dans une phrase explicite.  
4 resultat= (x+10)*2  
5 print(f"6) (x+10)*2 donne {resultat}.")  
6 print()  
7  
8 # 7. Effectuer l'addition de a et b, commenter le résultat.  
9 # Afficher les résultats dans une phrase explicite.  
10  
11 s=a+b  
12 print(f"7) La somme de a={a} et b={b} est s={s}. Les 3 premières puissances de 2.")  
13 print()  
14 # 8. Effectuer l'addition suivante : x+a ; commenter le résultat,  
15 # puis regarder le résultat de a+x.  
16  
17  
18 print(f"8) x={x} et a ={a} . Les tableaux ne sont pas de mme taille donc les opérations ne  
19 print()  
20  
21 # 9. Multiplier le tableau par le scalaire c que l'on fixera à 2.  
22 # Afficher les résultats dans une phrase explicite.  
23 resultat=x*2  
24 print(f"9) x*2 renvoie {resultat} .")  
25 print()  
26  
27 # 10. Effectuer la multiplication de a et b ; commenter le résultat.  
28 # Afficher les résultats dans une phrase explicite.  
29 p=a*b  
30 print(f"10) p=a*b renvoie p={p}. Multiplication terme à terme.")  
31 print()
```



```
1 # 11. Effectier la multiplication suivante : x*a ; commenter les résultats.
2 # Afficher les résultats dans une phrase explicite.
3
4 print(f"11) x={x} et a={a} n'ont pas les mêmes dimensions, donc la multiplication est impos
   .")
5 print()
6
7 # 12. Récupérer les positions des multiples de 2 et les stocker
8 # dans un tableau que l'on nommera ind,
9 # puis conserver uniquement les multiples de 2 de x dans un tableau
10 # que l'on nommera mult_2.
11 # Afficher les résultats dans une phrase explicite.
12 indice : list = []
13 for k in range(len(x)):
14     if x[k]%2==0 :
15         indice.append(k)
16 ind=np.array(indice)
17
18 mult_2=x[ind]
19 print(f"12) x={x} les indices des éléments pairs sont ind={ind} .")
20 print(f"    mult_2={mult_2}.")
21 print()
22 # 13. Afficher les éléments de x qui sont multiples de 3 et multiples de 2.
23 # Afficher les résultats dans une phrase explicite.
24 indice : list = []
25 for k in range(len(x)):
26     if x[k]%2==0 and x[k]%3==0:
27         indice.append(k)
28 ind=np.array(indice)
29
30 print(f"13) x ne contient pas de multiples de 2 et de 3 .")
31 print()
```

```
1 # 14. Afficher les éléments de x qui sont multiples de 3 ou multiples de 2.
2 # Afficher les résultats dans une phrase explicite.
3 indice : list = []
4 for k in range(len(x)):
5     if x[k]%2==0 or x[k]%3==0:
6         indice.append(k)
7 ind=np.array(indice)
8 print(f"14) x={x} les indices des éléments sont multiples de 2 ou 3
9     sont ind={ind} .")
9 print()
10 # 15. Calculer la somme des éléments de x.
11 # Afficher les résultats dans une phrase explicite.
12 somme : int = 0
13 for k in range(len(x)):
14     somme=somme + x[k]
15
16
17 print(f"14) x={x} , la somme de ses éléments vaut {somme}.")
18 print()
19
20 # 16. Remplacer le premier élément de x par un 4.
21 # Afficher les résultats dans une phrase explicite.
22 x[0]=4
23 print(f"16) On a x[0]={x[0]}.")
24 print()
```

```
1 # 17. Remplacer le premier élément de x par la valeur nan.
2 # Re-définir le tableau avec dtype=float,
3 # puis calculer la somme des éléments de x.
4 # Afficher les résultats dans une phrase explicite.
5 listeVal = [ 1, 2 ,3 ,4 ,5 ]
6 x=np.array(listeVal,dtype=float) # redéfinition du type des éléments du tableau pour y met
7 x[0]=np.nan
8 somme : int = 0
9 for k in range(len(x)):
10     somme=somme + x[k]
11 print(f"17) x={x} , la somme de ses éléments vaut {somme}.")
12 print()
13
14 # 18 Supprimer le tableau x. Le vérifier avec try-except.
15 # Afficher que le tableau n'existe plus dans une phrase explicite.
16
17 del x
18
19 # Vérification
20 try:
21     print(x)
22 except NameError:
23     print(f"18) une exception a été levée, le tableau x n'existe plus.")
24 print()
```