

**Exercice 1.** : Qu'obtient-on en exécutant ces codes ?

- Exemple a :

```
# Fonctions
def bonne_annee(l) :
    for e in l :
        print(e,end = " ")
# Main
bonne_annee(["je","vous","souhaite","une","bonne","annee"])
```

Donne :

- Exemple b :

```
# Fonctions
def dis_moi(l,code) :
    return l[0]+l[code]
# Main
message = dis_moi(["je t'aime ","pas du tout",
                  "un peu","passionnément"],1)
print(message)
message = dis_moi(["je t'aime ","pas du tout",
                  "un peu","passionnément"],0)
print(message)
```

Donne :

- Exemple c :

```
# Fonctions
def nsi_au_bac(liste_notes,liste_appreciation,i) :
    note = liste_notes[i]
    appreciation = liste_appreciation[i]
    return "note : "+ note + " , appréciation : " + appreciation
# Main
a = ["2/20" , "8/20" , "14/20" , "19/20"]
b = ["Nullissime" , "Normal" , "Très bien" , "Tricheur"]
resultat_bac_2022 = nsi_au_bac(a,b,len(a)-1)
print(resultat_bac_2022)
```

Donne :

- Exemple d :

```
# Fonctions
def tri(liste) :
    l = []
    for i in range(len(liste)) :
        if liste[i] > 10 :
            l.append(liste[i])
    return l

# Main
l = [18 , 4 , 9 , 14 , 1 , 13]
l_retour = tri(l)
print(l_retour)
print(tri([0,14,1,0,5,9,4,5,3,1,2,3]))
```

Donne :

- Exemple e :

```
# Fonctions
def bornes(l) :
    a = l[0]
    b = l[0]
    for i in range(len(l)) :
        if l[i] > a :
            a = l[i]
        if l[i] < b :
            b = l[i]
    return b , a

# Main
l = [18 , 4 , 9 , 14 , 1 , 13]
a_retour , b_retour = bornes(l)
print(a_retour , b_retour)
print(bornes([0,14,1,0,5,9,4,5,3,1,2,3]))
```

Donne :

**Exercice 2.:** Une liste de listes

On construit ci-dessous, la liste nommée *liste*. On exécute ce code et on exécute les instructions suivantes dans le *shell*. Que donnent-elles :

```
intitules = ["Prénom" , "Nom" , "birth day", "job actuel"]
l1 = ["Emmanuel" , "MACRON" , 1977 , "Président"]
l2 = ["Donald" , "TRUMP" , 1946 , "Ex-Président"]

liste = [intitules , l1 , l2 , ["zaza","WINDSOR",1926,"Queen"]]
```

<i>Instruction</i>	<i>Shell après exécution</i>
<code>&gt;&gt;&gt; liste[2]</code>	
<code>&gt;&gt;&gt; liste[2][2]</code>	
<code>&gt;&gt;&gt; len(liste)</code>	
<code>&gt;&gt;&gt; len(liste[1])</code>	
<code>&gt;&gt;&gt; len(liste[1][1])</code>	
<code>&gt;&gt;&gt; "MACRON"[2]</code>	
<code>&gt;&gt;&gt; len("MACRON")</code>	
<code>&gt;&gt;&gt; liste[1][1][2]</code>	
<code>&gt;&gt;&gt; "N" in "MACRON"</code>	
<code>&gt;&gt;&gt; "CO" in "MACRON"</code>	

**Exercice 3.**: Une chaîne de caractère peut être traitée comme une liste

La chaîne de caractère *mot* est définie dans un fichier par :

```
mot = "Bonjour les amis, je suis un humanoïde"
```

On exécute ce code et on exécute les instructions suivantes dans le *shell*. Que donnent-elles :

<i>Instruction</i>	<i>Shell après exécution</i>
<pre>&gt;&gt;&gt; len(mot)</pre>	
<pre>&gt;&gt;&gt; mot[5]</pre>	

⇒ On exécute le code suivant :

```
mot = "Bonjour les amis, je suis un humanoïde"

for i in range(len(mot)) :
    if i%2 == 0 : print(mot[i],end="")
```

Qu'obtient-on dans le shell ?