

I.P.E.I.M. Durée : 1 heure	Devoir surveillé n°2 – Informatique	Niveau : AP1 MP, PC, T Année Universitaire : 2020 – 2021
--------------------------------------	--	---

DOCUMENTS NON AUTORISÉS

**VOUS POUVEZ ÉVENTUELLEMENT UTILISER LES FONCTIONS PYTHON DECRITES A
L'ANNEXE**

Exercice 1. (4 pts)

On vous demande d'écrire une fonction récursive en Python qui prend en argument une chaîne de caractères et qui renvoie **True**; si cette chaîne est tautogramme et **False**; dans le cas contraire.

- **N.b:** Une chaîne est dite tautogramme si chacun de ses mots commence par la même lettre (sans aucune distinction entre majuscule et minuscule).
- **Exemple :** La chaîne " Racem rigole rapidement " est tautogramme.

Problème. (16 pts)

Les questions soulignées peuvent être considérées comme indépendantes.

'jj-mm-aaaa' et 'jj/mm/aaaa' sont des formats utilisés pour représenter une date avec huit chiffres et deux séparateurs ('-' ou '/') dont: les deux premiers chiffres représentent le jour, les deux chiffres suivants le mois et les quatre derniers chiffres l'année.

Exemple:

Le 12 mars 2021 est représenté dans ces formats par 12/03/2021 ou 12-03-2021.

Question 1:

Évaluer les commandes suivantes et compléter les réponses:

```
>>> s='12-03-2021'
```

```
>>> s.split()
```

```
.....
```

```
>>> s.split('-')
```

```
.....
```

```
>>> s.split('/')
```

```
.....
```

```
>>> s.replace('-', '/').split('/')
```

```
.....
```

Question 2:

Écrire une fonction *decouper* de paramètres une chaîne s et une séquence t et permet de retourner une liste de chaînes selon les éléments de la séquence t.

- Exemple:

```
>>> decouper('bonjour tout le monde','un')
['bo', 'jo', 'r to', 't le mo', 'de']
```

Question 3:

Écrire une fonction *liste_conv* qui prend en argument une liste de mots et renvoie la liste de ses éléments convertis en entier si c'est possible en cas des **exceptions**, elle doit retourner une liste vide [].

- Exemple:

```
liste_conv(['12', '03', '2021']) retourne [12, 03, 2021].
liste_conv(['abc', '03', '2021']) retourne [].
```

Question 4:

Soit 'mois', la variable globale suivante:

```
mois=[
    {'nb_j':31, 'libelle':'Janvier'},
    {'nb_j':28, 'libelle':'Février'},
    {'nb_j':31, 'libelle':'Mars'},
    {'nb_j':30, 'libelle':'Avril'},
    {'nb_j':31, 'libelle':'Mai'},
    {'nb_j':30, 'libelle':'Juin'},

    {'nb_j':31, 'libelle':'Juillet'},
    {'nb_j':31, 'libelle':'Août'},
    {'nb_j':30, 'libelle':'Septembre'},
    {'nb_j':31, 'libelle':'Octobre'},
    {'nb_j':30, 'libelle':'Novembre'},
    {'nb_j':31, 'libelle':'Décembre'}
]
```

Écrire une fonction *date_correcte* qui accepte une liste d et retourne **True** si la date est correcte **False** sinon.

- Exemple:

```
date_correcte([12, 10, 2020]) vaut True.
date_correcte([33, 10, 2020]) vaut False puisque le mois mm=10 qui est octobre ne contient pas
jj=33 jours.
```

Question 5:

En utilisant ce qui précède, écrire une fonction *saisir_date* qui permet de saisir une date correcte et de retourner sa liste détaillé des entiers [jj,mm,aaaa].

Question 6:

Écrire une fonction *max_dates* qui accepte deux dates d1 et d2 (sous ses formes détaillées) et retourne la date la plus récente.

Question 7:

a- Corriger les erreurs dans la fonction *diff_date* suivante: *d0* et *d1* sont supposés 2 listes.

```
def diff_date(d0,d1):
    years=d1(2)-d0(2)
    if (d1[1]>=d0[1])
        months=d1[1]-d0[1]br
    else
        months=12+d1[1]-d0[1]
        years=years-1
    if d1[0]>d0[0]:jours=d1[0]-d0[0]
    else:jours=mois[d1[0]]['nb_j']+d1[0]-d0[0]
    return {jour:jours,'month':month,'years':years}
```

b- Quel est le rôle de cette fonction?

Question 8 (Application):

En utilisant ce qui précède, écrire une fonction *verifier_anniv()* qui permet de:

- Saisir une date de naissance "*date_naissance*" sous la forme [jj,mm,aaaa].
- Saisir une date quelconque "*date_qque*" sous la forme [jj,mm,aaaa]. On exige évidemment que cette date soit plus récente que la *date_naissance*
- Afficher l'âge exacte.
- Vérifier si la *date_qque* correspond à son anniversaire et afficher un message de félicitation si oui.
- **Prendre en considération la gestion des exceptions.**

Annexe: Quelques méthodes sur les chaînes:

- `s.split(sep)` découpe la chaîne `s` suivant le séparateur `sep` et renvoie le résultat (ces strings) sous forme de liste. Si `sep=None`, les mots sont séparés par une suite quelconque d'espaces

Exemple:

```
>>>'a = b'.split('=')
['a ', ' b']
>>>'a b c d'.split()
['a', 'b', 'c', 'd']
```

- `s.replace(oldStr, newStr)` renvoie une nouvelle chaîne en remplaçant la chaîne `oldStr` (si elle existe) par une autre `newStr` dans `s`.

Exemple:

```
>>> s = 'a + b'
>>> ch = s.replace('+','*')
>>> ch
'a * b'
```

- `s.upper()` renvoie une chaîne où tous les caractères sont en majuscules. Les symboles et les nombres sont ignorés.

Exemple:

```
>>> s = 'Bonjour les amis'
>>> ch = s.upper()
>>> ch
'BONJOUR LES AMIS'
```

- `s.lower()` renvoie une chaîne où tous les caractères sont en minuscules. Les symboles et les nombres sont ignorés.

Exemple:

```
>>> s = 'Bonjour Ali'
>>> ch = s.lower()
>>> ch
'bonjour ali'
```