

## Examen de Langages de script n° 1 : 2011/2012

- 
- Durée : 2h
  - Vous devez éteindre et ranger vos téléphones.
  - Les programmes sont à faire en PYTHON 3.
  - **Chaque exercice doit être rédigée sur une copie distincte.**
  - L'annexe du sujet contient
    - des rappels de PYTHON ;
    - la documentation du module re.
- 

### Exercice 1 :

1. Écrire une fonction `filterdiv(l, i)` qui prend en argument une liste d'entiers `l` et un entier `i` strictement positif, et qui renvoie la sous-liste de `l` contenant les éléments de `l` qui sont divisibles par `i`.
2. Écrire une fonction `cprod(m, n)` qui prend deux nombres `n, m` en argument, et qui renvoie l'ensemble de tous les couples `(x, y)` formés d'un élément `x` de `m` et un élément `y` de `n`.
3. Écrire une fonction `listn(m, i)` qui prend un nombre `m` en argument, et renvoie l'ensemble de tous les listes d'éléments de `m` de longueur `i`.

4. Qu'imprime le programme suivant ?

```
d = {i:i*i for i in range(2,-3,-1)}
e = {}
for i in range(2,-3,-1):
    e[d[i]]=i
for i in range(-5,5):
    if i in d: print(d[i])
    if i in e: print(e[i])
```

5. Qu'imprime le programme suivant ?

```
def f(): print (); print('f'+chr(34)+'chr(34)')
f("def f(): print (); print('f'+chr(34)+'chr(34)')")
```

6. Qu'imprime le programme suivant ?

```
def shuffle(a, b, c = []):
    if a == []:
        return [c + b]
    if b == []:
        return [c + a]
    return shuffle(a[1:], b, c + [a[0]]) + shuffle(a, b[1:], c + [b[0]])
print(shuffle([1,2],[3,4]))
```

En général, qu'est-ce que fait la fonction `shuffle` appliquée à deux listes ?

7. Écrire une fonction qui étant donné un entier positif retourne un chaîne de caractères contenant la représentation de l'entier avec un virgule entre chaque groupe de 3 chiffres. Par exemple pour 1234, la fonction renvoie la chaîne de caractères "1,234", pour 3 elle renvoie "3" et pour 1234567 elle renvoie "1,234,567".

**Exercice 2 :**

L but d c t x rcie st d gér r d s SMS. Pour trait r l s qu stions qui suiv nt il st r commandé d lir l' x rcie n nti r t d'introduir tout s l s fonctions auxiliair s qu vous jug r z néc ssair n xplicant l ur rôl .

Un fichi r d SMS st un t xt d la form suivant :

```
0654342310 31/05/2012 10h23 S lut, t'es ou ?
0789304059 30/05/2012 11h10 C rl t' l issé un mess ge. R ppelle le : 0899432340
0654394503 01/06/2012 09h03 T' s p s l solution de l'exo 2 ?
0654394503 01/06/2012 09h40 Qu'est-ce que tu f is, y en m rre d' ttendre.
0660324524 29/05/2012 06h30 Rdv d ns 5 min u 0899 bvd Roy l.
0653434325 28/04/2012 03h20 Le nouve u IPHONE à 400$ chez phonehouse.
0143523523 03/06/2012 16h23 ADIDAS: PLUS QUE 200 MODELES DISPO CHEZ DECATHLON.
```

Chaqu lign conti nt xact m nt un SMS. L s numéros d téléphon ont tous 10 chiffr s, l s dat s sont toujours formé s d 10 caractèr s t l'h ur d 5. D s SMS s ront r prés ntés comm un dictionnair . L s cl fs sont l s numéros d téléphon d'origin t l s val urs d s list s d m ssag s. Chaqu m ssag st un list d 3 élém nts [dat ,h ur ,cont nu]. Choiss z un r prés ntation adéquat pour la dat t l'h ur .

On considèr qu'un SMS st publicitair s'il conti nt l caractèr \$ **ou** si plus qu la moitié d s caractèr s sont d s majuscul s. On considèr qu'un m ssag st un SPAM s'il conti nt un numéro d téléphon comm nçant av c 0899 (un numéro d téléphon a 10 chiffr s).

Pour répondr à c t x rcie , vous d v z rédig r d'un part un modul python nommé `sms.py` t d'autr part donn r l s xplications néc ssair s à la compréh nsion d votr cod .

1. Écri r un fonction qui étant donné comm argum nt l nom d'un fichi r d SMS r nvoi un dictionnair d SMS.
2. Écri r un fonction qui étant donné un dictionnair d SMS, r nvoi un dictionnair sans l s SMS SPAM.
3. Écri r un fonction qui élimin d'un dictionnair d SMS tous l s SMS publicitair s.
4. Écri r un fonction qui xtrait d'un dictionnair d SMS un "list noir " d numéros d téléphon qui ont nvoyé soit d s SMS publicitair s soit d s SMS SPAM.
5. Écri r un fonction qui p rm t d ch rch r un mot dans un dictionnair SMS. Ell doit r nvoy r tous l s numéros qui ont nvoyé un m ssag cont nant l mot. Att ntion : Si on ch rch "la" l d uxièm numéro n' st pas r nvoyé.
6. Écri r un fonction qui étant donné un dictionnair d SMS t un dat r nvoi un dictionnair d SMS sans l s SMS qui sont plus vi ux qu 2 s main s.
7. L modul `sms.py` s t rmin ra par la parti principal du cod qui n s ra xécuté qu lorsqu l modul st lancé n lign d command à partir d'un t rminal unix av c comm paramètr l nom d'un fichi r SMS. C tt parti principal imprim ra sur l'écran tous l s SMS non publicitair s.
8. Décri r l s opérations à ff ctu r afin qu la command `./sms` lanc l script corr spondant.

## Annexe

### a) Rappel de quelques éléments de PYTHON

- `range(i,j,l)` permet de parcourir les entiers de `i` à `j` exclu avec un pas de `l`.

```
>>> for i in range(3,-4,-1):  
...     print(i)  
...  
3  
2  
1  
0  
-1  
-2  
-3
```

- La fonction `chr` prend en argument un entier et renvoie le caractère correspondant dans l'encodage ASCII.

```
>>> print(chr(34))  
...  
...  
"
```

- Le point-virgule «`;`» permet de séparer plusieurs instructions s'enchaînant sur une même ligne.

```
>>> y = 5; x = 2*y; print(x)  
...  
...  
10
```

## b) Module re - descriptif basé sur le livre de Harold Erbin

**Syntaxe** L'expression répond à un syntaxe très codifié et possède de nombreux symbols ayant un sens particulier. Pour débuter, tout caractère alphanumérique n'a pas de signification spéciale : A correspond simplement à la lettre A, 1 au chiffre 1, etc. Quant aux principaux symbols spéciaux, il sont :

- . : désigne n'importe quel caractère ;
- ^ : indique que le début de la chaîne doit correspondre ;
- \$ : indique que la fin de la chaîne doit correspondre ;
- {n} : indique que le caractère précédant doit être répété n fois.
- {n,m} : indique que le caractère précédant doit être répété entre n et m fois.
- \* : le caractère précédant peut être répété aucun ou plusieurs fois. Par exemple, à `ab*` peut correspondre : a, ab, ou a suivi d'un nombre quelconque de b.
- + : le caractère précédant peut être répété un ou plusieurs fois. Par exemple, à `ab+` correspond un a suivi d'un nombre quelconque de b.
- ? : le caractère précédant peut être répété zéro ou un fois. Par exemple, à `ab?` correspond à `ab` ou `a`.

L'antislash permet d'échapper tous ces caractères spéciaux. Les crochets `[]` permettent d'indiquer un plage de caractère, par exemple `[-h]` correspondra à `,`, `f`, `g` ou `h`. Finalement, il existe quelques caractères spéciaux assez utiles :

- `\w` : il correspond à tout caractère alphabétique, c'est à dire qu'il est équivalent à `[a-zA-Z]` ;
- `\W` : il correspond à tout ce qui n'est pas un caractère alphabétique ;
- `\b` : il correspond à la frontière (début ou fin) d'un mot ;
- `\d` : il correspond à tout caractère numérique, c'est à dire qu'il est équivalent à `[0-9]` ;
- `\D` : il correspond à tout ce qui n'est pas un caractère numérique .

## Utilisation

`re.search(pattern, string)` Cherche le motif dans la chaîne passée en argument et retourne un `MatchObject` si des correspondances sont trouvées, sinon retourne `None`.

`re.split(pattern, string)` Découpe la chaîne `string` selon les occurrences du motif.

```
>>> re.split(r'\W', 'Truth is beautiful, without doubt.')
['Truth', 'is', 'beautiful', '', 'without', 'doubt', '']
```

`re.findall(pattern, string)` Retourne tous les sous-chaînes de `string` correspondant au motif.

`re.sub(pattern, repl, string)` Retourne la chaîne `string` où le motif a été remplacé par `repl`.