

Examen corrigé de LS4 - 2012 - session 1

Correction par Baptiste Fontaine pour IP7.

Exercice 1

Question 1

```
def filterdiv(l, i):  
    return [e for e in l if e%i == 0]
```

Question 2

```
def cprod(m, n):  
    return {(x, y) for x in m for y in n}
```

Question 3

```
def listn(m, i):  
    # on suppose que m contient des listes  
    return {e for e in m if len(e) == i}
```

Question 4

Le programme affiche les lignes suivantes:

```
4  
1  
0  
0  
1  
-1  
4  
-2
```

Question 5

Le programme affiche les lignes suivantes :

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

Question 6

Le programme affiche la liste de toutes les permutations de [1, 2, 3, 4].
La fonction `shuffle` appelée avec deux listes renvoie une liste de toutes les permutations de la concaténation de ces deux listes.

Question 7

```
def add_commas(n):  
    if n < 1000: # n a 3 chiffres ou moins  
        return str(n)  
  
    return add_commas(n/1000) + "," + str(n%1000)
```

Exercice 2

Question 1

```
def read_sms(name):  
    sms = {}  
    f = open(name)  
    for line in f.readlines():  
        els = line.strip().split(' ', 3)  
        num = els[0]  
  
        if num not in sms:  
            sms[num] = []  
  
        sms[num].append([els[1], els[2], els[3]])  
  
    f.close()  
    return sms
```

Question 2

Il faut ajouter `import re` en haut du fichier.

```
def is_spam(msg):  
    return re.search(r"\b0899\d{6}\b", msg)  
  
def no_spams(sms):  
    for num in sms:  
        sms[num] = [s for s in sms[num] if not is_spam(s[2])]
```

```
return sms
```

Question 3

```
def is_ad(msg):  
    return msg.find('$') >= 0 or len(re.findall(r'[A-Z]', msg)) > len(msg)/2  
  
def no_ads(sms):  
    for num in sms:  
        sms[num] = [s for s in sms[num] if not is_ad(s[2])]  
  
    return sms
```

Question 4

```
def blacklist(sms):  
    ls = []  
    for num in sms:  
        for s in sms[num]:  
            if is_spam(s[2]) or is_ad(s[2]):  
                ls.append(num)  
                break  
  
    return ls
```

Question 5

```
def search_sms(sms, word):  
    ls = []  
    for num in sms:  
        for s in sms[num]:  
            if re.search(r"\b" + word + r"\b", s[2]):  
                ls.append(num)  
                break  
  
    return ls
```

Question 6

```
def is_too_old(date):  
    # Cette fonction ne gère pas les années bissextiles ni les mois à 29/29 ou  
    # 31 jours, et la date d'aujourd'hui est codée en dur.
```

```

ds = date.split('/')
today = ['19', '05', '2014']

diff_years = int(today[2]) - int(ds[2])
diff_months = int(today[1]) - int(ds[1])
diff_days = int(today[0]) - int(ds[0])

return (diff_years*365 + diff_months*30 + diff_days) > 14

def remove_old(sms):
    for num in sms:
        sms[num] = [s for s in sms[num] if not is_too_old(s[0])]

    return sms

```

Question 7

On ajoute en bas du fichier le code suivant :

```

import sys

if __name__ == '__main__':
    if len(sys.argv) < 2:
        print("Usage: ./sms.py <fichier>")
    else:
        sms = no_ads(read_sms(sys.argv[1]))

        for num in sms:
            for s in sms[num]:
                d = s[0]
                h = s[1]
                m = s[2]
                print "SMS de " + num + ", recu le " + d + " à " + h + " : " + m

```

Question 8

Il faut ajouter la ligne suivante en haut du fichier :

```
#!/usr/bin/env python3
```

Et rendre le script exécutable:

```
chmod +x sms.py
```