1I001 — Éléments de programmation 1

Interrogation 3 – Groupe 11.1
Jeudi 27 novembre 2014

Nom :

Durée : 20 minutes — Aucun document autorisé hormis la carte de référence. Les réponses doivent **tenir dans l'encadré** prévu à cet effet.
L'interrogation porte surtout sur les compréhensions, vous devrez donc les utiliser autant que possible.

Exercice 1. À propos de liste des mots...

1. Écrivez une définition avec signature et hypothèses éventuelles de la fonction `moins_de` qui étant donné une liste de mots `L` et un entier naturel `n` retourne la liste des mots de moins de n caractères.

   Par exemple :

```
>>> moins_de(['des', 'mots', 'courts', 'et', 'pas', 'longs'],4)
['des', 'et', 'pas']
>>> moins_de(['Les', 'sanglots', 'longs', 'Des', 'violons', 'De', "l'", 'automne'],7)
['Les', 'longs', 'Des', 'De', "l'"]
```

```
def moins_de(L,n):
    """ list[str] * int -> list[str]
    hypothese: n >=0
    Retourne la liste des mots de moins de n caracteres"""

    return [ mot for mot in L if len(mot) < n ]
```

2. Écrivez une définition avec signature et hypothèses éventuelles de la fonction `analyse_lex` qui étant donné une chaine `c` et une liste de mots `L` retourne *True* si `c` contient un des éléments de `L`.

   Par exemple :

```
>>> analyse_lex("ai-je un espace ?",[' '])
True
>>> analyse_lex("ai-je des chiffres ?",['1','2','3','4','5','6','7','8','9','0'])
False
```

```
def analyse_lex(c,L):
    """ str * list[str] -> bool
    Retourne True si c contient un des elements de la liste L"""

    # char: str
    for char in c:
        if char in L:
            return True
    return False
```

3. Écrivez une définition avec signature et hypothèses éventuelles de la fonction `sans_voyelles` qui étant donné une liste de mots `L` retourne la liste des mots de plus de 2 caractères qui n'ont pas de voyelles.

Nom :

Rappel : les voyelles sont 'a','e','i','o','u' et 'y'.

Par exemple :

```
>>> sans_voyelles(['eh', 'pssst', '!', 'Brrr', 'il', 'fait', 'froid', 'non', '?'])
['pssst', 'Brrr']
```

```
def sans_voyelles(L):
    """ list[str] -> list[str]
    Retourne la liste des mots de plus de 2 caracteres
    sans voyelles de la liste L"""

    return [ mot for mot in L
             if not ( analyse_lex(mot,['a','e','i','o','u','y']) )
                 and len(mot) > 2 ]
```

Exercice 2. À propos de liste des nombres...

1. Écrivez une définition avec signature et hypothèses éventuelles de la fonction `liste_carre_multiples` qui étant donné deux entiers naturels non nuls `n` et `x` retourne la liste des `n` premiers multiples de `x` élevés au carré.

   Par exemple :

   ```
   >>> liste_carre_multiples(5,2)
   [4, 16]
   >>> liste_carre_multiples(12,3)
   [9, 36, 81, 144]
   ```

```
def liste_carre_multiples(n,x):
    """ int * int -> list[int]
    Retourne la liste du carre n premiers multiples de x"""

    return [ mult*mult for mult in range(1,n+1) if mult % x == 0 ]
```