

CONTROLE FLASH - PYTHON (45 minutes)

DATE : Novembre 2016

GROUPE 1

NOM :

PRENOM :

Q1. Complétez les lignes vides dans la session ci-dessous au toplevel.

```
>>> L= [[2*i,i] for i in range(3)]
>>> L

>>> L[1] = L[2] ; L[2] = L[1]
>>> L

>>> L[1][0] = 0
>>> L

>>> L[1] = 0
>>> L
```

Q2. Sans aucune **def**, rédigez un petit programme Python de 4 ou 5 lignes affichant la liste des **nombre premiers** inférieurs à 10000 et comportant *exactement* 3 fois le chiffre 7, ainsi que le plus grand d'entre eux. On suppose qu'il existe déjà une fonction **est_premier(n)** retournant True ou False suivant que n est premier ou pas.

Q3. Programmez une fonction **chiffres(n)** prenant un entier $n > 0$. Cette fonction retournera une liste d'entiers formée des chiffres décimaux de n. Exemple : `chiffres(12343) == [1, 2, 3, 4, 3]`

```
def chiffres(n) :
```

Q4. Avec une boucle **for**, programmez la fonction **remplace(x, y, L)** opérant par *mutation* de la liste L en remplaçant chaque apparition de y dans L par x (*x remplace y dans L*). Aucun résultat.

LIS = [5,8,4,6,8,8] ; `remplace(0,8,LIS)` ; `print(LIS)` \Rightarrow [5, 0, 4, 6, 0, 0]

```
def remplace(x,y,L) :
```

Q5. Nommons **ruban** une liste dont chaque élément est un tuple (op, arg) dans lequel op est l'un des mots 'F' (pour *forward*) ou 'L' (pour *left*) et où arg est un entier dans [-30,30]. Voici un exemple de ruban de longueur 6 :

[('F', -5), ('L', -17), ('F', -30), ('F', 12), ('L', -25), ('L', 1)]

Programmez une fonction **executer(r)** prenant un ruban quelconque r et demandant à une tortue d'exécuter les instructions du ruban r, par ex: `forward(-5)` suivi de `left(-17)`, etc. Cette fonction dessinera, puis à la fin retournera comme résultat la distance totale parcourue par la tortue.

```
def executer(r) :                    # et retourner la distance parcourue
```

Q6a. Les *polynômes de Valrose* V_n sont définis par récurrence de la manière suivante : $V_0 = 1$ et $V_n = X V_{n-1} + V'_{n-1}$. Par exemple $V_2 = X^2 + 1$. Calculez à la main l'expression mathématique du polynôme V_4 :

$V_4 =$

Q6b. Programmez la fonction **deriv(p)** qui retourne le polynôme *dérivé* du polynôme creux p. Utilisez-la ensuite pour définir la fonction **valrose(n)** retournant le polynôme creux V_n :

```
def deriv(p) :
```

```
def valrose(n) :
```