CONTROLE FLASH - PYTHON *(45 minutes)*

DATE : jeudi 24 novembre 2016 **SECTION : L1-Info** **GROUPE : 2**

NOM : **PRENOM :**

Question 1 : Programmez en une ligne de code une fonction `contient_chiffre(n, c)` qui retourne `True` si un nombre entier `n` contient le chiffre `c`, et `False` sinon.

```
def contient_chiffre(n, c) :

    return
```

```
>>> contient_chiffre(3241, 1)
True
>>> contient_chiffre(3241, 7)
False
```

Question 2a : La suite (U_n) est définie par récurrence de la manière suivante :

$$U_0 = 1,\ U_1 = 1,\ U_n = U_{n-1} + 2U_{n-2} \text{ pour } n \geq 2$$

Programmez une fonction `u(n)` récursive qui utilise directement la définition pour calculer et renvoyer U_n.

```
def u(n) :
```

Question 2b : La fonction précédente n'est pas très efficace. Par exemple pour calculer `u(5)` elle fait appel à `u(4)` et `u(3)`, `u(4)` fait lui-même appel à `u(3)` et `u(2)` et donc `u(3)` est calculé 2 fois. Utilisez un dictionnaire `memo` défini globalement pour réécrire la fonction `u` précédente en la rendant efficace (comme vu en cours et TP).

```
memo =
def u_memo(n) :
```

Question 3a : Un **système linéaire de deux équations à deux inconnues** x et y se présente sous la forme suivante, où a_1, b_1, c_1, a_2, b_2, c_2 sont les coefficients réels du système :

$$a_1x + b_1y = c_1$$
$$a_2x + b_2y = c_2$$

Un tel système sera mémorisé par **une liste de deux listes**. La première de ces listes mémorise les coefficients a_1, b_1, c_1 et la deuxième les coefficients a_2, b_2, c_2.
Ecrivez en Python la liste `s` qui mémorise le système :

$$x - 2y = 0$$
$$2x + 3y = 1$$

```
s =
```

CONSEIL : pour les deux questions suivantes, écrivez sur le papier la correspondance entre les coefficients et les éléments de `s`, a_1 : `s[0][0]`, b_1 : `s[0][1]`, etc.

Question 3b : Un tel système possède une solution unique si et seulement si son déterminant $a_1b_2 - a_2b_1$ est non nul. Programmez une fonction `determinant(s)` qui renvoie le déterminant du système mémorisé par la liste `s` comme indiqué précédemment.

```
def determinant(s) :
```

Question 3c : Si le déterminant du système est non nul, il possède alors un unique couple solution donné par les formules de Cramer :

$$x = (c_1b_2 - c_2b_1) / det$$
$$y = (a_1c_2 - a_2c_1) / det$$

où det est le déterminant du système.
Programmez une fonction `resoud(s)` qui renvoie le couple solution quand le déterminant du système mémorisé par la liste `s` n'est pas nul, `False` sinon.

```
def resoud(s) :
```

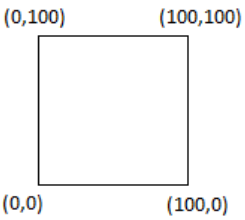
```
>>> resoud([[1,1,-1], [2,2,7]])
False
>>> resoud([[1,-1,-1], [2,2,7]])
(1.25, 2.25)
```

Question 4 : D'après la Wikipédia : « En mathématiques, une ligne polygonale, ou ligne brisée est une figure géométrique formée d'une suite de segments, la seconde extrémité de chacun d'entre eux étant la première du suivant. Un **polygone** est une ligne polygonale fermée (la seconde extrémité du dernier segment est égale à la première extrémité du premier segment) ». On considère ici que les extrémités des segments sont des points du plan et l'on s'intéresse aux polygones. On suppose qu'un polygone est mémorisé par une liste de points, chaque point étant mémorisé par un **couple** de ses coordonnées. On vous demande de programmer une fonction `dessine_polygone(p)` qui dessine avec la tortue le polygone défini par sa liste de points `p`.

```
def dessine_polygone(p) :
```

Par exemple, dessine_polygone([(0,0), (100,0), (100,100), (0,100)])

donne le résultat ci-contre :



Question 5 : Le cours propose deux représentations pour les polynômes : la représentation *creuse* et la représentation *dense*. Indiquez succintement les contextes pour lesquels l'une est plus efficace que l'autre et réciproquement.