


<b>I.P.E.I.Sfax</b>  <b>A.U : 2018/2019</b>	<b>Devoir : Semestre 2</b> <b>Matière : Informatique</b> <b>Classes : 1ère année MP, PC et PT</b> <b>Durée : 1H</b>	
---	--	---

### Exercice 1 (Récursivité croisée)

Ecrire trois fonctions récursives qui calculent respectivement les termes  $u_n$ ,  $v_n$  et  $w_n$  des suites  $u$ ,  $v$ ,  $w$  définies par:

$$u_0 = 1, v_0 = 2, w_0 = 3 \text{ et } \forall n \in \mathbb{N}$$

$$\bullet u_{n+1} = \frac{2}{3}u_n + \frac{1}{6}v_n + \frac{1}{6}w_n$$

$$\bullet v_{n+1} = \frac{1}{6}u_n + \frac{2}{3}v_n + \frac{1}{6}w_n$$

$$\bullet w_{n+1} = \frac{1}{6}u_n + \frac{1}{6}v_n + \frac{2}{3}w_n$$

### Exercice 2

La numérotation romaine utilise 2 règles :

(1) Toute lettre  $L_2$  placée à la droite d'une autre lettre  $L_1$  s'ajoute à celle-ci si  $L_2 \leq L_1$

Exemple : VI = 5 + 1 = 6 et XX = 10 + 10 = 20. Ainsi, si un chiffre à une valeur supérieure ou égale à celle du chiffre suivant alors on additionne à la valeur de tout le reste.

(2) Toute lettre d'unité  $L_1=I$  placée immédiatement à la gauche d'une autre lettre  $L_2 \neq I$

se retranche de celle-ci. Ainsi, si un chiffre à une valeur inférieure à celle du chiffre suivant alors on le soustrait de la valeur de tout le reste.

Exemple : IV = 5 - 1 = 4 et IX = 10 - 1 = 9

La règle (2) est élargie à toute lettre  $L_1$  placée immédiatement à la gauche d'une autre lettre  $L_2$ :

si  $L_2 > L_1$  alors  $L_1$  retranche de  $L_2$ .

Exemple : XC = 100 - 10 = 90

Exemple :

1980 en chiffres romains MCMLXXX      1981 en chiffres romains MCMLXXXI

2000 en chiffres romains MM      2001 en chiffres romains MMI

Ecrire une fonction `chiffres_romains_to_int` qui convertit une chaîne de caractères représentant l'écriture en chiffres romains d'un entier naturel  $n$  en sa représentation en base 10 (type `int`).

Indications : on utilise le dictionnaire ci-dessous

`dict_rom = {"M":1000, "D":500, "C":100, "L":50, "X":10, "V":5, "I":1}`

### Exercice 3

La fonction qui calcule les valeurs de la série de Fibonacci est définie par :

- $u_0=0$
- $u_1=1$
- $u_n = u_{n-1} + u_{n-2}$

Écrire cette fonction sous forme itérative et sous forme récursive. Laquelle des deux variantes est préférable en terme de mémoire ?

### Exercice 4

En cryptographie, le chiffrement par décalage, aussi connu comme le chiffre de César ou le code de César, est une méthode de chiffrement très simple utilisée par Jules César dans ses correspondances secrètes.

Le texte chiffré s'obtient en remplaçant chaque lettre du texte clair original par une lettre à distance fixe, toujours du même côté, dans l'ordre de l'alphabet. Pour les dernières lettres (dans le cas d'un décalage à droite), on reprend au début. Par exemple avec un décalage de 3 vers la droite, A est remplacé par D, B devient E, et ainsi de suite jusqu'à W qui devient Z, puis X devient A etc. Il s'agit d'une permutation circulaire de l'alphabet.

#### Exemple :

Si **CH**='ABCCZABEY' alors la chaîne **Res** crypté qui sera affichée est 'BCDDABCEZ'

- 1) Ecrire une fonction `Crypto_cesar(ch)` ayant comme entrée une chaîne **CH** et comme sortie sa chaîne crypté selon le principe décrit. On suppose qu'on aura à traiter uniquement les caractères de l'alphabet en minuscule: a, b, ...,z. Tout les autres caractères de ponctuation ou chiffres ne subiront aucun changement

Pour cela vous devrez utiliser la fonction prédéfinie `ord` qui renvoie le nombre entier représentant le code Unicode du caractère représenté par la chaîne donnée. Par exemple, `ord('a')` renvoie le nombre entier 97, `ord('b')`=98, ...etc. Il s'agit de l'inverse de `chr()`.

- 2) Ecrire une fonction `DeCrypto_cesar(ch)` ayant comme entrée une chaîne **CCH** et comme sortie sa chaîne décrypté selon le principe inverse de ce qui a été décrit. On suppose qu'on aura à traiter uniquement les caractères de l'alphabet en minuscule: a, b, ...,z. Tout les autres caractères de ponctuation ou chiffres ne subiront aucun changement