CENTRE UNIVERSITAIRE JEAN-FRANÇOIS CHAMPOLLION – ALBI

L1 Maths-Info-PCI **Python impératif** **Vendredi 20 décembre 2013**

CONTRÔLE N°3 (*1 h 30 min*)

Les exercices sont indépendants. Rendre 5 fichiers `exo1_toto.py`, `exo2_toto.py`, *etc. où* `toto` *désignera votre nom (sans accent ni espace). À la fin de l'épreuve, zipper vos fichiers en un fichier de nom* `toto_cc3.zip` *où* `toto` *désignera votre nom, sans accent ni espace.*

Précisions : **a)** Lorsque des contraintes portant sur les paramètres d'une fonction sont imposées par l'énoncé, il n'est pas demandé à la fonction de vérifier que les paramètres vérifient ces contraintes. **b)** Tous les exemples donnés dans les énoncés doivent être testés.

Exercice 1

Ecrire une procédure $f(n)$ où $n \geq 1$ est un entier et qui affiche les entiers de 1 à n avec les contraintes suivantes :

- si le nombre à afficher est multiple de 3, au lieu de l'afficher, la procédure affiche le mot **coco** ;
- de même, si le nombre est multiple de 4, la procédure affiche seulement le mot **rico** ;
- enfin, si le nombre est multiple, à la fois, de 3 et de 4, la procédure affichera uniquement le mot **cocorico**.

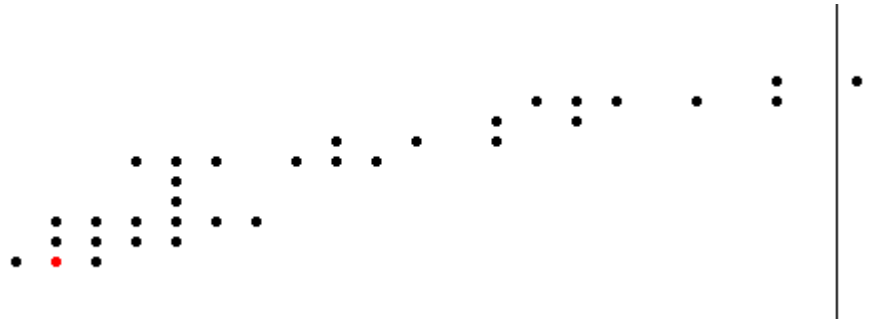
Les affichages seront effectués sur des lignes distinctes. Par exemple, l'appel $f(13)$ doit afficher les lignes ci-contre.

L'évaluation de cet exercice ne tiendra pas seulement compte de l'affichage produit mais aussi de la qualité du code.

```
1
2
coco
rico
5
coco
7
rico
coco
10
11
cocorico
13
```

Exercice 2

Vous allez devoir simuler, sur une aire de dessin Turtle, les mouvements d'une grenouille qui s'entraîne à la course. La grenouille démarre du point de coordonnées $(-200, 0)$ et doit franchir, après une succession de sauts, la ligne d'arrivée qui est la droite verticale $x = 190$. La grenouille se déplace aléatoirement en effectuant des sauts de 40 vers la droite ou bien des sauts de 20 vers la gauche ou encore des sauts de 10 vers le haut. Elle ne se déplace jamais vers le bas. Vous devez dessiner un point rouge au point de départ, tracer la ligne d'arrivée (choisissez vous-même sa longueur) et simuler les déplacements de la grenouille. Comme la grenouille se déplace par bonds, vous devez seulement dessiner un petit disque noir aux endroits où la grenouille attérit. Votre programme doit s'arrêter après le premier bond qui franchit la ligne d'arrivée. Vous n'êtes pas obligé de créer une fonction. Une simulation pourrait s'afficher comme ci-dessous :



Exercice 3

Ecrire une fonction $f(L, M)$ qui prend deux listes d'entiers en paramètres et retourne True si les deux listes sont «opposées» et False sinon. Deux listes sont considérées comme «opposées» si elles ont le même nombre d'éléments et si, à des indices identiques, elles possèdent des éléments opposés (comme -81 et 81). Voici quelques exemples de comportements de la fonction f :

L	[81, -12, 0, -81, -31]	[-81]	[0, 0]	[ ]	[81, -12]	[81, -12]	[-81, 12, 0]
M	[-81, 12, 0, 81, 31]	[81]	[0, 0]	[ ]	[-81, -12]	[-81, 12, 0]	[81, -12]
$f(L, M)$	True	True	True	True	False	False	False

T.S.V.P.

Exercice 4

(1) Ecrire une fonction $g(n)$ qui prend en paramètre un entier $n \geq 0$ et renvoie le multiple de 5 le plus proche de n. Le tableau ci-dessous donne quelques exemples du comportement de g.

n	42	64	15	90	0
$g(n)$	40	65	15	90	0

(2) Vous allez devoir écrire une fonction f destinée à arrondir une heure donnée aux 5 minutes les plus proches. Voici plusieurs exemples d'arrondis d'heures :

Heure	14h53	18h31	2h10	1h02	9h58	23h58	23h57
Heure arrondie	14h55	18h30	2h10	1h00	10h00	00h00	23h55

Une heure de la journée sera codée par deux nombres entiers h et m, où h est le nombre d'heures ($0 \leq h < 24$) et m le nombre de minutes ($0 \leq m < 60$). Par exemple, 14 h 05 est codée par $h = 14$ et $m = 5$ ou encore 4 h sera codée par $h = 4$ et $m = 0$.

Écrire une fonction f qui prend deux paramètres h et m représentant une heure donnée et qui renvoie l'heure arrondie à 5 minutes près. Plus précisément, la fonction f devra retourner la liste $[H, M]$ correspondant à l'heure arrondie. Voici un tableau montrant quelques exemples de comportements de la fonction f :

Heure exacte	h	m	Heure arrondie $f(h, m)$
14h 53	14	53	[14, 55]
18h 31	18	31	[18, 30]
2h 10	2	10	[2, 10]
1h 02	1	2	[1, 0]
9h 58	9	58	[10, 0]
23h 58	23	58	[0, 0]
23h 57	23	57	[23, 55]

On fera attention de ne pas écrire des nombres de minutes ou d'heures avec un zéro initial (comme 02), Python 3 considérant cette syntaxe comme une erreur.

Exercice 5

(1) Soit, par exemple, la liste L d'entiers : [3, 7, 5, 5, 8, 8, 8, 8, 2, 5, 5, 5]. Elle contient des répétitions *successives* comme 8, 8, 8, 8 ou encore 5, 5, 5. La liste sans répétition successive est [3, 7, 5, 8, 2, 5].

Plus généralement, écrire une fonction $f(L)$, où L est une liste non vide d'entiers, qui renvoie la liste M formée des éléments de L sans ses répétitions de termes successifs. La liste M doit être une *nouvelle* liste et la liste L ne doit pas être modifiée par la création de M. Voici quelques exemples de listes L et de listes $f(L)$:

Liste L	Liste $f(L)$ sans répétitions successives
[3, 7, 5, 5, 8, 8, 8, 8, 2, 5, 5, 5]	[3, 7, 5, 8, 2, 5]
[7, 7, 7]	[7]
[7, 0, 1]	[7, 0, 1]
[7]	[7]

(2) (*Bonus*) Ecrire une fonction $g(L)$, où L est une liste d'entiers, qui renvoie la longueur maximale d'un bloc d'éléments successifs de L et qui sont égaux. Voici quelques exemples du comportement de g :

L	$g(L)$	Commentaire
[7, 7, 5, 5, 8, 8, 8, 8, 5, 5, 5]	4	Le bloc 8, 8, 8, 8 est de longueur 4
[7, 7, 7]	3	Le bloc 7, 7, 7 est de longueur 3
[7, 0, 1]	1	Aucune répétition dans L