Nom: Prénom: Groupe:

Université de Bordeaux — Devoir surveillé du 10 novembre 2017
Département Licence, **4TPM101U** — 14h30-16h

Le sujet comporte 5 exercices et 7 pages, dont une page d'annexe

Aucun document n'est autorisé – Toutes les fonctions de manipulation d'images disponibles sont rappelées en annexe page 7. Vous pouvez détacher cette annexe pour plus de facilité.

Exercice 1 Considérons la fonction `mystere` suivante prenant en paramètre une liste de nombres `L` et un nombre `x` :

```
def mystere(L, x):
    cpt = 0
    for i in L :
        if i > x:
            cpt = cpt + 1
            if cpt == 3:
                return True
        else:
            cpt = 0
    return False

juilletTemperaturesMax = [19,23,25,31,34,32,35,29,26,24,26,22,
   23,26,28,31,34,37,26,24,22,23,24,23,23,21,23,26,30,26,25]

temperatureCaniculeCalvados = 30
temperatureCaniculeGironde = 35
```

1. Simulez l'exécution de l'appel
   `mystere(juilletTemperaturesMax, temperatureCaniculeCalvados)`
   en complétant le tableau suivant montrant l'évolution des variables `i` et `cpt` :

i		
cpt		

2. Que retourne l'appel
   `mystere(juilletTemperaturesMax, temperatureCaniculeCalvados)` ?

3. Que retourne l'appel
   `mystere(juilletTemperaturesMax, temperatureCaniculeGironde)` ?

4. Quelle condition (nécessaire et suffisante) doivent vérifier la liste `L` et la valeur `x` pour que l'appel `mystere(L, x)` retourne la valeur `True`.

Exercice 2 Météo

1. Écrire le code de la fonction `aGele(ListeTemperatures)` qui a pour paramètre une liste de nombres `ListeTemperatures` et qui retourne `True` s'il existe une valeur strictement négative dans cette liste, `False` sinon.

   Par exemple `aGele([1, 0, 2, 8, 6, 9])` retourne `False` alors que `aGele([1, 0, -2, 8, 6, 9])` retourne `True`.

2. Écrire le code de la fonction `aPluTousLesJours(ListePrecipitations)` qui a pour paramètre une liste de nombres `ListePrecipitations` et qui retourne `True` si toutes les valeurs de cette liste sont strictement supérieures à zéro, `False` sinon.

   Par exemple `aPluTousLesJours([1, 0, 20, 80, 60, 90])` retourne `False` alors que `aPluTousLesJours([8, 30, 20, 23, 16, 19])` retourne `True`.

Nom: Prénom: Groupe:

Exercice 3 Nombres parfaits

(Chaque question de cet exercice peut-être traitée de manière indépendante des autres : vous pouvez utiliser les fonctions des questions précédentes, même si vous n'avez pas réussi à les implémenter).

1. Écrire une fonction `estDiviseur(i,n)` qui retourne `True` si `i` est un diviseur de `n` et `False` sinon.

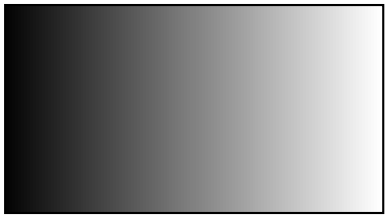
2. Écrire une fonction `sommeDiviseurs(n)` qui retourne la somme des diviseurs de `n` strictement plus petits que lui.

3. On dit qu'un nombre est *parfait* s'il est égal à la somme de ses diviseurs strictement plus petits que lui. Par exemple, 6 est un nombre parfait car 6=1+2+3. Les nombres 15 et 28 sont-ils parfaits ? Justifiez vos réponses.

4. Écrire une fonction `estParfait(n)` qui retourne `True` si `n` est un nombre parfait, `False` sinon.

Exercice 4 Dégradés

1. Écrivez une fonction `degradeHorizontal(img)` qui dessine dans l'image `img` un dégradé de gris horizontal allant du noir à gauche au blanc à droite. Pour chaque pixel de coordonnée `(x,y)`, le niveau de gris correspondant est donné par la formule : `256 * x // img.width`. L'image résultante doit ressembler à celle-ci (sans le cadre autour qui ne sert ici qu'à vous montrer le bord de l'image) :



2. **Question à traiter en dernier !**
   Écrivez une fonction `degradeDiagonale(img)` qui dessine dans l'image `img` un dégradé de gris allant du noir depuis le coin en haut à gauche jusqu'au blanc en bas à droite. L'image résultante doit ressembler à celle-ci :

Nom: Prénom: Groupe:

Exercice 5 Pixels foncés

On dispose d'une image en niveau de gris, c'est à dire que tous les pixels ont une couleur RGB de la forme `(c,c,c)`. On ne souhaite conserver que les pixels les plus foncés et rendre les autre pixels blancs.

1. Écrire une fonction `pixelsFonces(img,seuil)` qui modifie l'image `img` en remplaçant chaque pixel de couleur `(c,c,c)` par un pixel blanc lorsque `c > seuil` et en laissant le pixel tel quel dans le cas contraire.

2. Écrire la suite d'instructions qui
   (a) ouvre le fichier `chienchien.png` contenant l'image à gauche ci-dessus et la stocke dans une variable ;
   (b) appelle la fonction `pixelsFonces` sur cette image en utilisant un seuil de 72 ;
   (c) affiche l'image obtenue.

Nom: Prénom: Groupe:

Annexe : voici un rappel des principales fonctions disponibles pour manipuler les images (à vous de voir celles qui sont utiles pour ce devoir).

L'argument `img` est une image	
`open(nom)`	Ouvre le fichier *nom* et retourne l'image contenue dedans (par exemple `open("teapot.png")`.
`Image.save(img, nom)`	Sauvegarde l'image *img* dans le fichier *nom*.
`new("RGB",(largeur,hauteur))`	Retourne une image de taille $largeur \times hauteur$, initialement noire.
`largeur = img.width` `hauteur = img.height`	Récupère la largeur de *img*. Récupère la hauteur de *img*.
`Image.putpixel(img, (x,y), (r,g,b))`	Peint le pixel (x, y) dans l'image *img* de la couleur (r, g, b)
`(r,g,b) = Image.getpixel(img, (x,y))`	Retourne la couleur du pixel (x, y) dans l'image *img*