

NOM :

Prénom :

Classe : PSI*

Devoir Surveillé 3

Les réponses sont à écrire exclusivement sur ce document

Les différentes questions sont indépendantes les unes des autres.

1 Un problème de rendu de monnaie

On s'intéresse ici au nombre de manière différentes de rendre une somme n en fonction d'une liste de type de pièces disponible stockées dans une liste L . On considérera que l'on dispose de suffisamment de pièces de chaque type pour rendre la somme n .

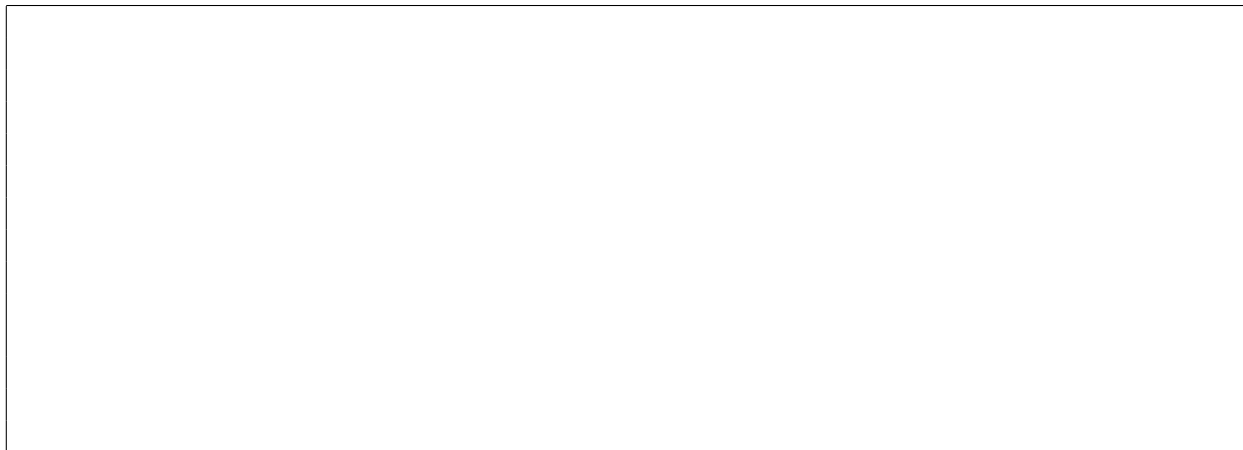
On va écrire une fonction `pieces(n,L)` qui va renvoyer le nombre de manières différentes d'obtenir n avec des pièces ayant des valeurs dans L .

Question 1. Montrer que `pieces(3,[1,2])` vaut 2
Que vaut `pieces(2,[0.5,1])` ?

Question 2. Compléter le programme suivant :

```
def pieces(n,L):  
    if n==0:  
        return  
    if L=[] or n<0:  
        return  
    else:  
        M=L[0]  
        return
```

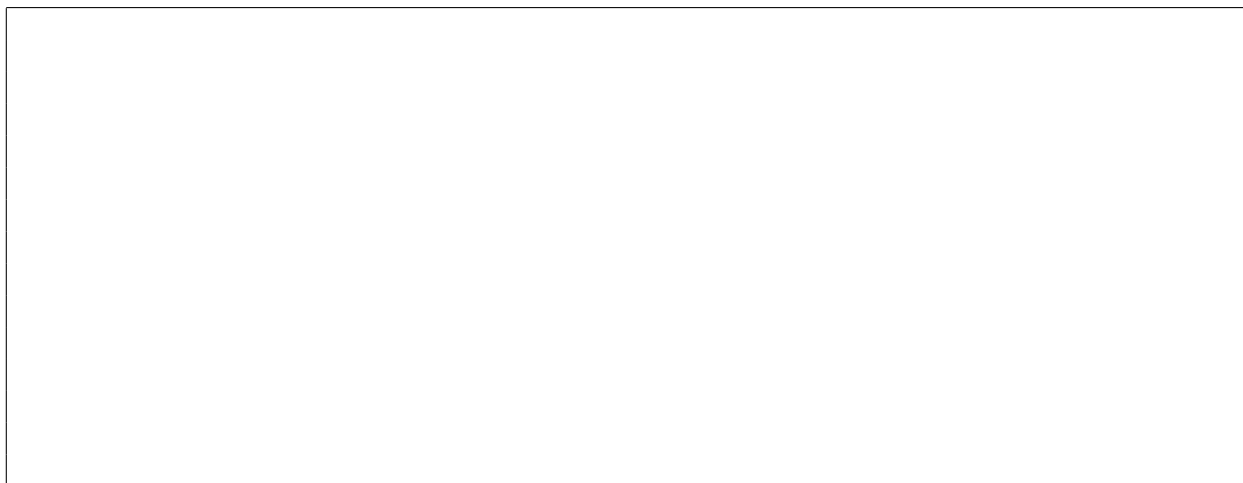
Question 3. Montrer que ce programme se termine.



2 Transformations de listes récursives

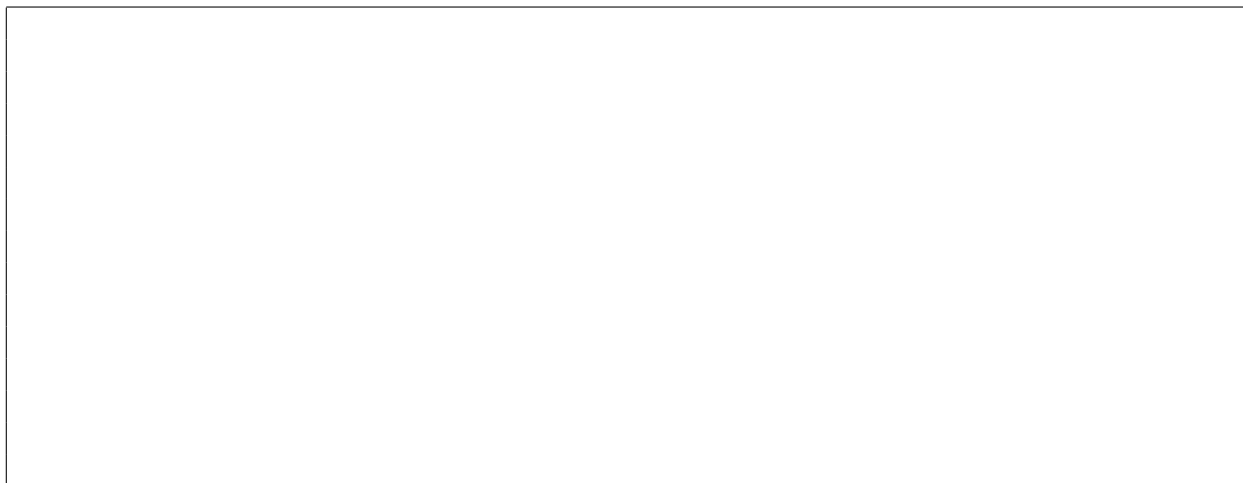
Question 4. Écrire une fonction récursive `swap2` qui prend en argument une liste `L` et qui renvoie cette liste où l'ordre des éléments est modifié de sorte à ce que cette liste contienne : le second élément de `L` puis le premier, puis le quatrième, puis le troisième, puis le sixième, puis le cinquième etc...

Ainsi `swap2([1,4,9,16,25,36,49])` renverra `[4,1,16,9,36,25,49]`



Question 5. Réécrire cette fonction dans un cas plus général : écrire une fonction récursive `swap` qui prend en arguments une liste `L` et un entier strictement positif `k` qui renvoie la liste constituée du $k^{\text{ème}}$ élément de `L`, puis du $k - 1^{\text{ème}}$, ..., puis du premier, puis du $2k^{\text{ème}}$... du $k + 1^{\text{ème}}$ etc...

Ainsi `swap([1,4,9,16,25,36,49],3)` renverra `[9,4,1,36,25,16,49]`



Question 6. On considère dans cette question que l'affectation ou l'ajout d'un élément à une liste a un coût de 1, et qu'un slicing de liste a un coût égal à la longueur de la liste slicée : ainsi $L[a:b]$ a un coût de $b - a$ et $L[:]$ a un coût de $n = \text{len}(L)$

Exprimer et justifier la complexité (asymptotique) de l'algorithme écrit précédemment en fonction de n et de k . Si besoin, on pourra considérer que n est grand devant k .

3 Algèbre linéaire et numpy

Cet exercice sera traité en utilisant le type `array` du module `numpy`

Question 7. Écrire une fonction `transpose` qui prend en argument une matrice M et renvoie la matrice transposée M^T .

Question 8. Écrire une fonction `test` qui prend en argument un couple de matrices (A, M) , renvoie un message d'erreur si A et M ne sont pas des matrices carrées de même taille et la matrice $AM + MA^T$ sinon.

Question 9. Écrire une fonction `base_canonique` d'argument un entier naturel n et qui renvoie une liste dont les éléments sont, dans l'ordre usuel, les éléments de la base canonique de $\mathcal{M}_n(\mathbb{R})$.

Question 10. Écrire une suite d'instructions définissant la matrice $A = \begin{pmatrix} 1 & 2 & -3 \\ 4 & 2 & 0 \\ 5 & 6 & 1 \end{pmatrix}$ et générant la matrice de l'application $M \mapsto AM + MA^T$ dans la base canonique de $\mathcal{M}_n(\mathbb{R})$.

4 Traitement d'images

Une image en couleur acquise par une caméra est constituée de trois couches (RGB).

Les données de l'image sont stockées dans un tableau à trois dimensions : la première dimension correspond à la couleur (rouge, vert ou bleu, indice 0, 1 ou 2), la seconde correspond à la coordonnée selon x et la troisième à la coordonnée selon y . Ainsi, les dimensions du tableau sont : $3 \times m \times n$ où $m = 800$ et $n = 600$.

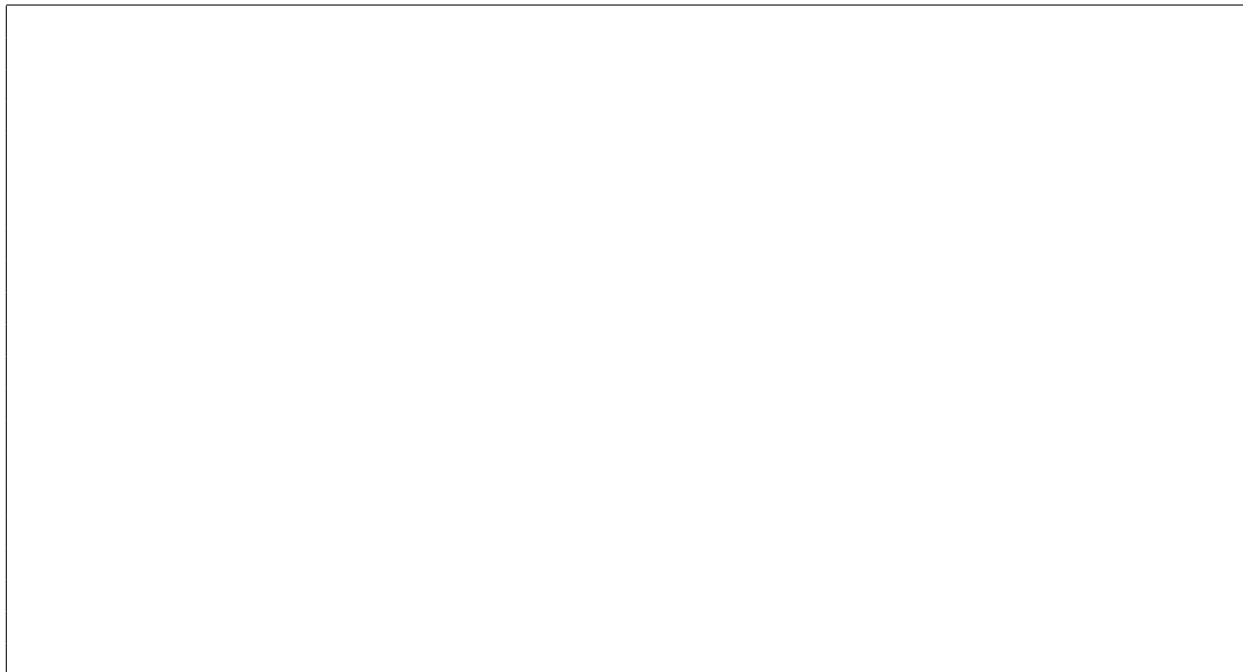
La valeur associée à chaque pixel est un entier compris entre 0 et 255.

Question 11. Donner la quantité de mémoire nécessaire en octets pour stocker le tableau représentant l'image émise par la caméra en justifiant le codage retenu pour un pixel d'une couche.

Question 12. Dans le cadre du traitement des images acquises, la première étape est de convertir l'image en couleur en niveaux de gris. La méthode consiste à rechercher le maximum et le minimum pour chaque pixel sur les trois couches, puis à faire la moyenne de ces maxima et minima et ainsi obtenir la valeur du nouveau pixel en niveaux de gris.

On note `imagecolor` le tableau représentant une image en couleur.

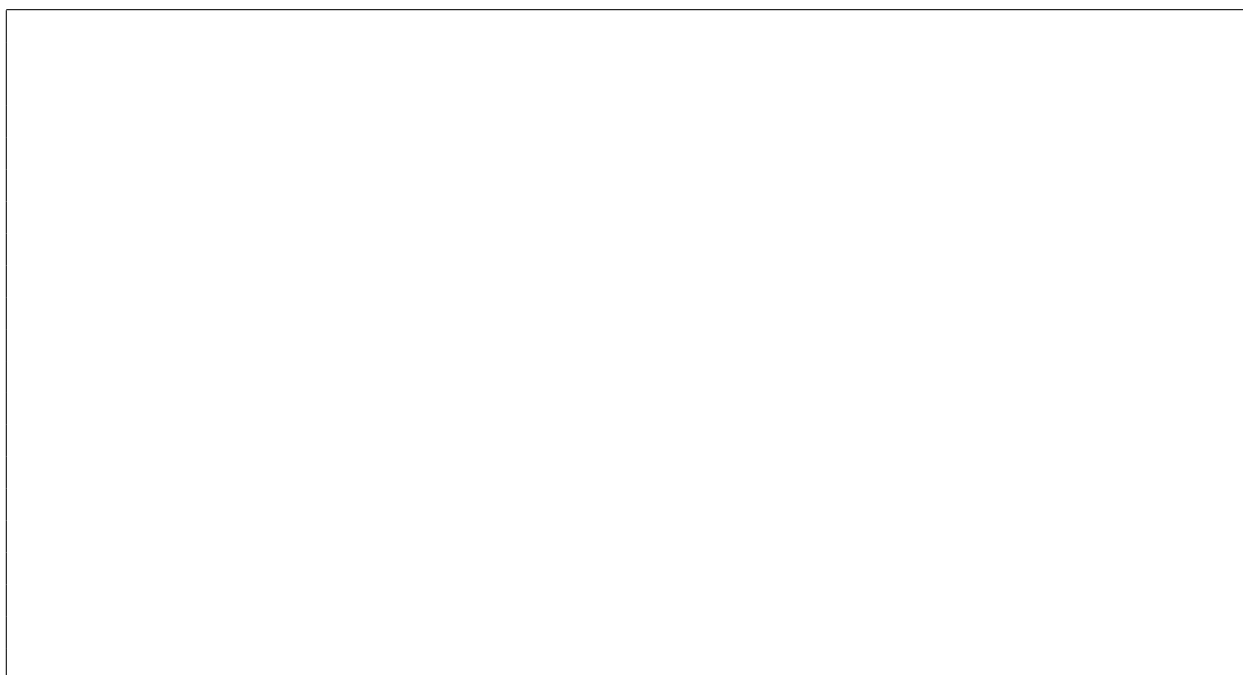
Écrire une fonction `grayscale(imagecolor)` qui renvoie une image en niveaux de gris (qui sera notée `image` dans l'algorithme principal), sous forme de tableau à deux dimensions de taille $m \times n$ contenant des valeurs entières comprises entre 0 et 255, en suivant l'algorithme décrit ci-dessus. Il est possible d'utiliser les fonctions `min` et `max` de Python.



5 Bijections de \mathbb{N} vers \mathbb{N}^2

On construit une bijection de \mathbb{N} sur \mathbb{N}^2 en numérotant les points entiers du quart de plan supérieur droit diagonale par diagonale et de bas en haut : ainsi, les premiers éléments sont $u_0 = (0, 0)$, $u_1 = (1, 0)$, $u_2 = (0, 1)$, $u_3 = (2, 0)$, $u_4 = (1, 1)$, $u_5 = (0, 2)$, $u_6 = (3, 0)$, etc...

Question 13. Écrire une fonction récursive prenant en argument (a, b) et calculant l'indice n tel que $u_n = (a, b)$.



Question 14. Coder également la fonction réciproque, c'est-à-dire la fonction qui à n associe u_n . On commencera par montrer que, pour $T_p = \frac{p(p+1)}{2}$, si $u_n = (x_n, y_n)$ et $T_p \leq n < T_{p+1}$, alors $x_n + y_n = T_p$.