

Vous avez à disposition une vidéo ainsi que les images associées à cette vidéo. Cette vidéo est l'acquisition d'un pendule vert oscillant sur un fond blanc.

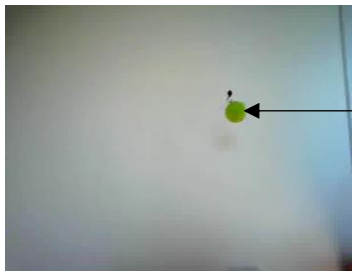
On rappelle les éléments suivants :

```
from PIL import Image
import numpy as np

im=Image.open("output_00205.jpg")#ouverture de l'image n°205
tab=np.array(im)#création d'un tableau tab associé à l'image n°205
nouvelle_image = Image.fromarray(tab)#création d'une image de tab
nouvelle_image.save("photo.jpg")#enregistrement au format .jpg
nouvelle_image.show()# affichage de la photo créée
```

a) Critère de détection du pendule sur une image

Le pendule est de couleur verte et oscille sur un fond blanc.



Pour cette photo, le pendule est approximativement à la 200^e ligne et la 430^e colonne.

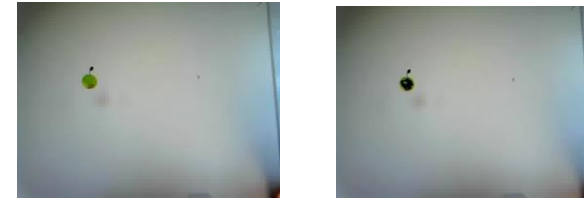
```
print(tab[195,429])#--->[ 81 132  0]
print(tab[200,429])#--->[ 93 127  0]
print(tab[205,429])#--->[103 122  0]
print(tab[195,435])#--->[ 89 141  0]
print(tab[195,425])#--->[ 81 132  0]
```

La couleur verte est associée à un niveau de bleu nul (ou quasi-nul). Dans la suite, nous supposons que la présence du pendule se traduit par une composante bleue inférieure à 5.

- 1) A l'aide d'un `mask`, proposer une fonction prenant pour argument le tableau numpy d'une photo et retournant une photo où les pixels occupés par le pendule sont colorés en noir.

b) Détermination du barycentre du pendule sur une image

Le pendule est représenté par un grand nombre de pixels. On note N le nombre de pixels aux points M_i qui ont été colorés en noir après application du critère précédent.



Pour identifier la position du pendule, nous utiliserons le barycentre G des N pixels noirs défini tel que :

$$\overrightarrow{OG} = \frac{1}{N} \sum_{i=1}^N \overrightarrow{OM_i} = \frac{\sum_{i=1}^N x_i}{N} \overrightarrow{u_x} + \frac{\sum_{i=1}^N y_i}{N} \overrightarrow{u_y} = x_G \overrightarrow{u_x} + y_G \overrightarrow{u_y}$$

- 2) En s'inspirant de la fonction précédente, proposer une fonction `barycentre(im)` admettant une image `im` comme argument et renvoyant les coordonnées (x_G, y_G) en pixels du point.

Aide : `np.around(a)` : renvoie la valeur entière la plus proche du float `a`

`np.meshgrid(x, y)` : renvoie deux tableaux de dimension $(x*y)$, un contenant toutes les valeurs de `x` et l'autre de `y`

`np.mean(a)` : renvoie la valeur moyenne

c) Tracé de la trajectoire du pendule

Tracer la trajectoire du pendule en traitant toutes les photos.

Rq :

```
for i in range(1,242):
    file = ("output_{:05d}.jpg".format(i))
    #notation permettant d'avoir output_0000i
```