

Corrigé du TP Informatique 21

Exercice 1

On saisit :

```
def negatif(tab):  
    """negatif(tab:list)->list  
    Renvoie la liste de listes de points  
    de l'image decrite par tab en negatif"""  
    L,C=len(tab),len(tab[0])  
    return [[255-tab[i][j] for j in range(C)] for i in range(L)]
```

Exercice 2

1. Le plus simple est d'utiliser des listes par compréhension :

```
def assombrir(tab,s):  
    """negatif(tab:list)->list  
    Renvoie la liste de listes de points  
    de l'image decrite par tab assombrie d'un facteur s"""  
    L,C=len(tab),len(tab[0])  
    return [[int(s*tab[i][j]) for j in range(C)] for i in range(L)]
```

2. On utilise deux fois la fonction negatif :

```
def eclaircir(tab,s):  
    """negatif(tab:list)->list  
    Renvoie la liste de listes de points  
    de l'image decrite par tab eclaircie d'un facteur s"""  
    return negatif(assombrir(negatif(tab),s))
```

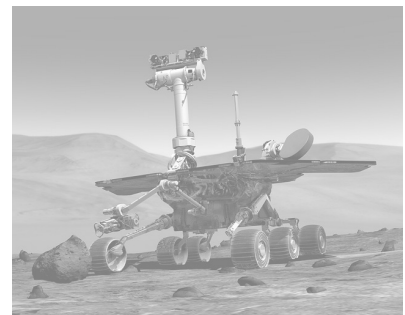
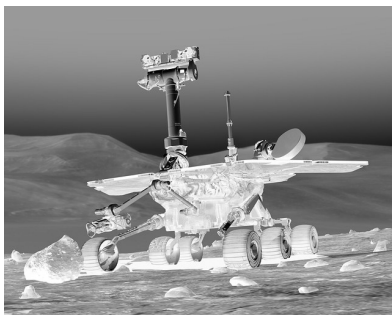


FIGURE 1 – Négatif, image assombrie et éclaircie

Exercice 3

```
def homothetie(tab,mu):
    """homothetie(tab:list,mu:float)->list
    Réalise l'homothétie de facteur mu sur l'image
    décrite par la liste de listes de points tab"""
    L,C=len(tab),len(tab[0])
    mL,mC=int(mu*L),int(mu*C)
    res=[[0]*mC for k in range(mL)]
    i0,j0=L//2,C//2
    I0,J0=mL//2,mC//2
    def xy_ij(x,y):
        return int(i0-y),int(j0+x)
    def IJ_XY(I,J):
        return J-J0,I0-I
    for I in range(mL):
        for J in range(mC):
            X,Y=IJ_XY(I,J)
            x,y=1/mu*X,1/mu*Y
            i,j=xy_ij(x,y)
            res[I][J]=tab[i][j]
    return res
```

Exercice 4

```
def rotation(tab,a):
    """rotation(tab:list,a:float)->list
    Réalise la rotation d'angle a sur l'image
    décrite par la liste de listes de points tab"""
    L,C=len(tab),len(tab[0])
    i0,j0=L//2,C//2
    def xy_ij(x,y):
        return int(i0-y),int(j0+x)
    def ij_xy(i,j):
        return j-j0,i0-i
    def deborde(i,j):
        return i<0 or i>=L or j<0 or j>=C
    res=[[0]*C for k in range(L)]
    ca,sa=np.cos(a),np.sin(a)
    for I in range(L):
        for J in range(C):
            X,Y=ij_xy(I,J)
            x,y=ca*X+sa*Y,-sa*X+ca*Y
            i,j=xy_ij(x,y)
            if not deborde(i,j):
                res[I][J]=tab[i][j]
    return res
```