TD I

Prise en main

1 Fonctions

Question 2.4

```
def moyenne_1(a,b):
    return (a+b)/2
```

Question 2.5

```
def moyenne_pondere(a,coef_a,b,coef_b):
    c=(a*coef_a+b*coef_b)
    d=(coef_a+coef_b)
    return c/d
```

Question 2.6

```
def moyenne_2(a,b):
    return moyenne_pondere(a,1,b,1)
```

Question 2.7

```
def pair(n):
    return n\%2==0
```

2 Conditionnelles

Question 3.1

```
def compare(a,b):
    if a<b:
        return -1
    if a==b:
        return 0
    if a>b:
        return 1
```

Question 3.2

```
def max2(x,y):
    if compare(x,y)==-1:
        return y
    if compare(x,y)==1:
        return x
    if compare(x,y)==0:
        return x,y
```

Question 3.3

```
def max3(x,y,z):
    return(max2(z,max2(x,y)))
```

3 Suppléments

Question 4.1

```
def coutPhotocopies(n):
    if n<=10:
        total=0.20*n
    if 10<n<=30:
        total=0.20*10+0.15*(n-10)
    if n>30:
        total=0.20*10+0.15*20+0.10*(n-30)
    return total
```

Question 4.2

```
def uneMinuteEnPlus(x,y):
    if y==59:
        y=-1
        x=x+1
    y=y+1
    if x==24:
        x=0
    return(x,y)
```

Question 4.3

```
def solve(a,b,c):
    delta=b**2-4*a*c
    if delta==0:
        print("Une seule solution réelle:")
        return -b/(2*a)
    if delta>0:
        print("Deux solutions réelles:")
        return (-b+delta**0.5)/2*a,(-b-delta**0.5
```

TD II
Listes

2 Boucles for

Question 2.5

```
def factorielle(n):
    factorielle=1
    for i in range(1,n+1):
        factorielle=s*i
    return factorielle
```

Question 2.6

```
def factorielle2(L):
    s=1
    for i in L:
        print("i=",i)
        print(factorielle(i))
```

Question 2.7

```
def sommeImpairs(k):
    s=0
    for i in range(1,2*k,2):
        s=s+i
    return s
```

Question 2.8

```
def sommeBizarre(n):
    s=0
    for i in range(1,n+1):
        i=(3*i+1)*(3*i+1)
        s=s+i
    return s
```

Question 2.9

```
    s=1
    for p in range(0,k):
        p= 2**s
        s=s+p
    return s
```

3 Exploration de listes

Question 3.1

```
def moyenne(L):
    n=0
    for p in L:
        n=n+p
    return n/len(L)
```

Question 3.2

```
def nbApparitions (x,L):
    n=0
    for i in L:
        if i==x:
            n=n+1
    return n
```

Question 3.3

```
def doublons(L):
    for x in L:
        if nbApparitions(x,L)>1:
            return True
    return False
```

Question 3.4

```
def maxDesImpairs(L):
    n=0
    for p in L:
        if p\%2==1 and p>n:
            n=p
    if n==0:
        print("pas d'impairs")
    return n
```

Question 3.5

```
def minDesImpairs(L):
    s=[]
    for i in L:
        if i\%2==1:
            s=s+[i]
    if s==[]:
        return ('no impairs')
    return min(s)
```

Question 3.6

```
def dernierPlusGrand(L):
    m = moyenne(L)
    s=[]
    for i in range(len(L)):
        if L[i]>m:
            s=s+[i]
    return max(s)
```

Question 3.7

```
def plusproche(L):
    m = moyenne(L)
    n= abs(m-L[0])
    for i in L:
        if abs(m-i)<n:
            n=abs(m-i)
            r=i
    return i
```

4 Suppléments

Question 4.1

```
def tri(L):
    for i in range(len(L)):
        k=i
        for j in range(i+1,len(L)):
            if L[k]>L[j]:
                k=j
        L[k],L[i]=L[i],L[k]
    return L
def mediane(L):
    n=len(L)
    l=tri(L)
    if n%2==0:
        return (l[(n-1)//2]+l[(n+1)//2])//2
    else:
        return l[(n-1)//2]
```

Question 4.2

```
def PremierTour(L):
    for x in L:
        if nbApparitions(x,L) >len(L)/2:
            return x
```

Question 4.3

```
def PremierTour_2(L):
    S=[]
    for x in L:
        if S==[]:
            S=S+[x]
        if S!=[]:
            if S[0]==x:
                S=S+[x]
            if S[0]!=x:
                del S[0]
    S.reverse()
    if S!=[]:
        return S[0]
    return 0
```

TD III
Images

2 Pour commencer

Question 2.2

```
def rectangleCreux(img,x1,x2,y1,y2,c):
    (x1,x2,y1,y2)=ordre(x1,x2,y1,y2)
    (l,h)=img.size
    x2=min(x2,l-1)
    y2=min(y2,h-1)
    for x in range(x1,x2+1):
        Image.putpixel(img,(x,y1),c)
        Image.putpixel(img,(x,y2),c)
    for y in range(y1,y2+1):
        Image.putpixel(img,(x1,y),c)
        Image.putpixel(img,(x2,y),c)
    Image.show(img)
```

Question 2.3

```
def diagonale(img):
    (l,h)=img.size
    if l>h:
        for x in range(l):
            Image.putpixel(img,(x,x*h//l),(255,255,255))
    else:
        for y in range(h):
            Image.putpixel(img,(y*l//h,y),(255,255,255))
    Image.show(img)
```

3 Manipulation de couleurs

Question 3.1

```
def filtreRouge(img):
    img_rouge=img
    (l,h)=img.size
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img_rouge,(x,y))
            Image.putpixel(img,(x,y),(r,0,0))
    Image.show(img_rouge)
def filtreVert(img):
    (l,h)=img.size
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img,(x,y))
            Image.putpixel(img,(x,y),(0,g,0))
    Image.show(img)
def filtreBleu(img):
    img_bleu=img
    (l,h)=img.size
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img_bleu,(x,y))
            Image.putpixel(img_bleu,(x,y),(0,0,b))
    Image.show(img_bleu)
```

Question 3.2

```
def luminosite(img,facteur):
    f=int(facteur)
    (l,h)=img.size
    for x in range(l):
        for y in range (h):
            (r,b,g)=Image.getpixel(img,(x,y))
            Image.putpixel(img,(x,y),(r*f,g*f,b*f))
```

Question 3.3

```
def monocrhome(img):
    (l,h)=img.size
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img,(x,y))
            g=(r+g+b)//3
            Image.putpixel(img,(x,y),(g,g,g))
```

Question 3.4

```
def flou(img) :
    (l,h)=img.size
    nimg=new("RGB",(l,h),(0,0,0))
    for x in range(1,l-3):
        for y in range(1,h-3):
            (R,G,B)=(0,0,0)
            for i in range(-2,3):
                for j in range(-2,3):
                    (r,g,b)=Image.getpixel(img,(x+i,y+j))
                    R=r+R
                    G=g+G
                    B=b+B
            R=R//9
            B=B//9
            G=G//9
            Image.putpixel(nimg,(x,y),(R,G,B))
    return nimg
```

4 Agrandissement d'une image

Question 4.1

```
def agrandirfacteur2(img):
    (l,h)=img.size
    nimg=new("RGB",(l*2,h*2))
    for x in range(l):
        for y in range(h):
                (r,g,b)=Image.getpixel(img,(x,y))
                Image.putpixel(nimg,(2*x,2*y),(r,g,b))
                Image.putpixel(nimg,(2*x+1,2*y+1),(r,g,b))
                Image.putpixel(nimg,(2*x,2*y+1),(r,g,b))
                Image.putpixel(nimg,(2*x+1,2*y),(r,g,b))
    return nimg
```

5 Suppléments

Question 5.1

```
def gris(img):
    (l,h)=img.size
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img,(x,y))
            g=(r+g+b)//3
            Image.putpixel(img,(x,y),(g,g,g))

def noir_blanc(img):
    gris(img)
    (l,h)=img.size
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img,(x,y))
            if r>100:
                Image.putpixel(img,(x,y),(0,0,0))
            else:
                Image.putpixel(img,(x,y),(255,255,255))
```

Question 5.2

```
def floydSteinberg(img):
    (l,h)=img.size
    for y in range(1,h-1):
        for x in range(1,l-1):
            (ar,ag,ab)=Image.getpixel(img,(x,y))
            g=(ar+ag+ab)//3
            Image.putpixel(img,(x,y),(g,g,g))
            (dr,dg,db)=(ar-g,ag-g,ab-g)
            (r_1,g_1,b_1)=Image.getpixel(img,(x+1,y))
            Image.putpixel(img,(x+1,y),(r_1+(7//16)*dr,g_1+(7//16)*dg,b_1+(7//16)*db))
            (r_2,g_2,b_2)=Image.getpixel(img,(x-1,y+1))
            Image.putpixel(img,(x-1,y+1),(r_2+(3//16)*dr,g_2+(3//16)*dg,b_2+(3//16)*db))
            (r_3,g_3,b_3)=Image.getpixel(img,(x,y+1))
            Image.putpixel(img,(x,y+1),(r_3+(5//16)*dr,g_3+(5//16)*dg,b_3+(5//16)*db))
            (r_4,g_4,b_4)=Image.getpixel(img,(x+1,y+1))
            Image.putpixel(img,(x+1,y+1),(r_4+(1//16)*dr,g_4+(1//16)*dg,b_4+(1//16)*db))
```

Question 5.3

```
def cercle2(img,a,b,r):
    x=0
    y=r
    m=5-4*r
    while x<=y:
        if m>0:
            y=y-1
            m=m-8*y
        x=x+1
        m=m+8*x+4
        Image.putpixel(img,(x+a,y+b),(255,255,255))
        Image.putpixel(img,(y+a,x+b),(255,255,255))
        Image.putpixel(img,(-x+a,y+b),(255,255,255))
        Image.putpixel(img,(-y+a,x+b),(255,255,255))
        Image.putpixel(img,(x+a,-y+b),(255,255,255))
        Image.putpixel(img,(y+a,-x+b),(255,255,255))
        Image.putpixel(img,(-x+a,-y+b),(255,255,255))
        Image.putpixel(img,(-y+a,-x+b),(255,255,255))
```

TD IV
Modification d'image

1 Ouvrons le placard

Question 1.1

```
def openplacard(img):
    (l,h)=fantome.size
    for x in range(71,132):
        for y in range(4,187):
            (r,g,b)=Image.getpixel(img,(x,y))
            Image.putpixel(img,(x-64,y),(r,g,b))
            if x<l and y<h:
                (R,G,B)=Image.getpixel(fantome,(x-71,y-4))
                Image.putpixel(img,(x,y),(R,G,B))
            else:
                Image.putpixel(img,(x,y),(0,0,0))
```

Question 1.2

```
def openplacard_2(img):
    (l,h)=fantome.size
    for x in range(187,135,-1):
        for y in range(4,187):
            (r,g,b)=Image.getpixel(img,(x,y))
            Image.putpixel(img,(x+32,y),(r,g,b))
            if x-135<l and y<h:
                (R,G,B)=Image.getpixel(fantome,(x-68,y-4))
                Image.putpixel(img,(x,y),(R,G,B))
            else:
                Image.putpixel(img,(x,y),(0,0,0))
```

2 Niveaux de couleur

Question 2.1

```
def minimum(img):
    (l,h)=img.size
    (r,g,b)=Image.getpixel(img,(0,0))
    for x in range(l):
        for y in range(h):
            (rl,gl,bl)=Image.getpixel(img,(x,y))
            if rl<r:
                r=rl
            if gl<g:
                g=gl
            if bl<b:
                b=bl
    return r,g,b

def maximum(img):
    (l,h)=img.size
    (r,g,b)=Image.getpixel(img,(0,0))
    for x in range(l):
        for y in range(h):
            (rf,gf,bf)=Image.getpixel(img,(x,y))
            if rf>r:
                r=rf
            if gf>g:
```

```
                g=gf
            if bf>b:
                b=bf
    return r,g,b
```

Question 2.2

```
def LvlCouleurs(img):
    (l,h)=img.size
    (r_M,g_M,b_M)=maximum(img)
    (r_m,g_m,b_m)=minimum(img)
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img,(x,y))
            r_n=255*(r-r_m)//(r_M-r_m)
            g_n=255*(g-g_m)//(g_M-g_m)
            b_n=255*(b-b_m)//(b_M-b_m)
            Image.putpixel(img,(x,y),(r_n,g_n,b_n))
```

3 Détection de contours

Question 3.1

```
def contour_V(img):
    (l,h)=img.size
    nimg=new("RGB",(l,h))
    for x in range(1,l-1):
        for y in range(1,h):
            (r,g,b)=Image.getpixel(img,(x,y))
            (r2,g2,b2)=Image.getpixel(img,(x-1,y))
            (r3,g3,b3)=Image.getpixel(img,(x+1,y))
            Image.putpixel(nimg,(x,y),(2*r-r2-r3,2*g-g2-g3,2*b-b2-b3))
    return nimg
```

Question 3.2

```
def contour_H(img):
    (l,h)=img.size
    nimg=new("RGB",(l,h))
    for x in range(1,l):
        for y in range(1,h-1):
            (r,g,b)=Image.getpixel(img,(x,y))
            (r2,g2,b2)=Image.getpixel(img,(x,y-1))
            (r3,g3,b3)=Image.getpixel(img,(x,y+1))
            Image.putpixel(nimg,(x,y),(2*r-r2-r3,2*g-g2-g3,2*b-b2-b3))
    return nimg
```

Question 3.3

```
def norme(img):
    (l,h)=img.size
    nimg=new("RGB",(l,h))
    for x in range(1,l-1):
        for y in range(1,h-1):
            (r,g,b)=Image.getpixel(img,(x,y))
            (r2,g2,b2)=Image.getpixel(img,(x-1,y))
            (r3,g3,b3)=Image.getpixel(img,(x+1,y))
            (r4,g4,b4)=Image.getpixel(img,(x,y-1))
            (r5,g5,b5)=Image.getpixel(img,(x,y+1))
            Image.putpixel(nimg,(x,y),(4*r-r2-r3-r4-r5,4*g-g2-g3-g4-g5,4*b-b2-b3-b4-b5))
    return nimg
```

4 Symétries

Question 4.1

```
def miroir(img):
    (l,h)=img.size
    nimg=new("RGB",(l,h))
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img,(x,y))
            Image.putpixel(nimg,(l-x-1,y),(r,g,b))
    return nimg
```

Question 4.2

```
def retourner(img):
    (l,h)=img.size
    nimg=new("RGB",(l,h))
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img,(x,y))
            Image.putpixel(nimg,(l-x-1,h-y-1),(r,g,b))
    return nimg
```

5 Crypter une image

Question 5.1

```
def bloc(img,x1,x2,y1,y2):
    (l,h)=img.size
    R=0
    G=0
    B=0
    nbr=len(list(range(x1,x2)))*len(list(range(y1,y2)))
    for x in range(x1,x2):
        for y in range(y1,y2):
            (r,g,b)=Image.getpixel(img,(x,y))
            R=R+r
            G=G+g
            B=B+b
    for x in range(x1,x2):
        for y in range(y1,y2):
            Image.putpixel(img,(x,y),(R//nbr,G//nbr,B//nbr))
```

Question 5.2

```
def newimg(img):
    (l,h)=img.size
    nimg=new("RGB",(l,h))
    for x in range(l):
        for y in range(h):
            (r,g,b)=Image.getpixel(img,(x,y))
            Image.putpixel(nimg,(x,y),(r,g,b))
    return nimg

def crypte(img,k):
    (l,h)=img.size
    nimg=newimg(img)
    for x in range(0,l+1,k):
        for y in range(0,h+1,k):
            a=x+k
            b=y+k
```

```
            if a>l:
                a=l
            if b>h:
                b=h
            bloc(nimg,x,a,y,b)
    return nimg
```

TD V
Images Avancées

1 Une image peut en cacher une autre

Question 1.1

```
def valeur_dernier_bits_2(n):
    return n%8
```

Question 1.2

```
def decaler_2(n):
    return n*32+16
```

Question 1.3

```
def devoilerimage(img):
    (l,h)=img.size
    cible=new("RGB",(l,h))
    for x in range(l):
        for y in range (h):
            (r,g,b)=Image.getpixel(img,(x,y))
            Image.putpixel(cible,(x,y),(decaler_2(valeur_dernier_bits_2(r)),decaler_2(valeu
    return(cible)
```

Question 1.4

```
def premiers5(R,G,B):
    RL=[]
    GL=[]
    BL=[]
    for x in range(7,-1,-1):
        if R-2**x>=0:
            RL=RL+[1]
            R=R-2**x
        else:
            RL=RL+[0]
        if G-2**x>=0:
            GL=GL+[1]
            G=G-2**x
        else:
            GL=GL+[0]
        if B-2**x>=0:
            BL=BL+[1]
            B=B-2**x
        else:
            BL=BL+[0]
    return [RL[0],RL[1],RL[2],RL[3],RL[4]],[GL[0],GL[1],GL[2],GL[3],GL[4]],[BL[0],BL[1],BL

def premiers3(r,g,b):
    rl=[]
    gl=[]
    bl=[]
    for x in range(7,-1,-1):
        if r-2**x>=0:
            rl=rl+[1]
            r=r-2**x
        else:
            rl=rl+[0]
```

```
        if g-2**x>=0:
            gl=gl+[1]
            g=g-2**x
        else:
            gl=gl+[0]
        if b-2**x>=0:
            bl=bl+[1]
            b=b-2**x
        else:
            bl=bl+[0]
    return [rl[0],rl[1],rl[2]],[gl[0],gl[1],gl[2]],[bl[0],bl[1],bl[2]]

def recomposer(R,G,B,r,g,b):
    (RL,GL,BL)=premiers5(R,G,B)
    (rl,gl,bl)=premiers3(r,g,b)
    RL=RL+[rl[0]]+[rl[1]]+[rl[2]]
    GL=GL+[gl[0]]+[gl[1]]+[gl[2]]
    BL=BL+[bl[0]]+[bl[1]]+[bl[2]]
    (nbr_RL,nbr_GL,nbr_BL)=(0,0,0)
    for x in range(0,8):
        decimal_RL=RL[x]*2**(7-x)
        nbr_RL= nbr_RL+decimal_RL
        decimal_GL=GL[x]*2**(7-x)
        nbr_GL= nbr_GL+decimal_GL
        decimal_BL=BL[x]*2**(7-x)
        nbr_BL= nbr_BL+decimal_BL
    return (nbr_RL,nbr_GL,nbr_BL)

def dissimuler_image(img1,img2):
    (l,h)=min(img1.size,img2.size)
    nimg=new("RGB",(l,h))
    for x in range(l):
        for y in range(h):
            (R,G,B)=Image.getpixel(img1,(x,y))
            (r,g,b)=Image.getpixel(img2,(x,y))
            (rn,gn,bn)=recomposer(R,G,B,r,g,b)
            Image.putpixel(nimg,(x,y),(rn,gn,bn))
    return nimg
```

2 Fusion et bannière

Question 2.1

```
def fusion_lineaire(img1,img2):
    (l,h)=min(img1.size,img2.size)
    nimg=new("RGB", (l,h))
    for y in range(h):
        for x in range(l):
            (r_1,g_1,b_1)=Image.getpixel(img1,(x,y))
            (r_2,g_2,b_2)=Image.getpixel(img2,(x,y))
            r=(y*r_1+(h-y)*r_2)//h
            g=(y*g_1+(h-y)*g_2)//h
            b=(y*b_1+(h-y)*b_2)//h
            Image.putpixel(nimg,(x,y),(r,g,b))
    return nimg
```

Question 2.2

```
def fusion_quadratique(img1,img2):
    (l,h)=min(img1.size,img2.size)
    nimg=new("RGB", (l,h))
```

```
    for y in range(h):
        for x in range(l):
            (r_1,g_1,b_1)=Image.getpixel(img1,(x,y))
            (r_2,g_2,b_2)=Image.getpixel(img2,(x,y))
            r=(y**2*r_1+(1-y**2)*r_2)//(y**2+(1-y)**2)
            g=(y**2*g_1+(1-y**2)*g_2)//(y**2+(1-y)**2)
            b=(y**2*b_1+(1-y**2)*b_2)//(y**2+(1-y)**2)
            Image.putpixel(nimg,(x,y),(r,g,b))
    return nimg
```

Question 2.3

```
def banière(img1,img2):
    (l,h)=min(img1.size,img2.size)
    nimg=new("RGB",(l,h))
    lineaire=fusion_lineaire(img1,img2)
    quadra=fusion_quadratique(img1,img2)
    for x in range(0,1,4):
        for y in range(h):
            (r1,g1,b1)=Image.getpixel(img1,(x,y))
            (r2,g2,b2)=Image.getpixel(img2,(x,y))
            (rl,gl,bl)=Image.getpixel(lineaire,(x,y))
            (rq,gq,bq)=Image.getpixel(quadra,(x,y))
            Image.putpixel(nimg,(x,y),(r1,g1,b1))
            Image.putpixel(nimg,(x+1,y),(r2,g2,b2))
            Image.putpixel(nimg,(x+2,y),(rl,gl,bl))
            Image.putpixel(nimg,(x+3,y),(rq,gq,bq))
    return nimg
```

3 Taguer une image

Question 3.1

```
def insertionPossible(img1,img2,x,y):
    (L,H)=img1.size
    (l,h)=img2.size
    if x+l>L:
        return False
    return True
```

Question 3.2

```
def insereTexteNoir(img1,img2,x,y):
    if insertionPossible(img1,img2,x,y)==True:
        (L,H)=img1.size
        (l,h)=img2.size
        for i in range (l):
            for j in range (h):
                (r,g,b)=Image.getpixel(img2,(i,j))
                Image.putpixel(img1,(x+i,y+j),(r,g,b))
    else:
        print("Impossible")
```

Question 3.3

```
def insereTexteSombre(img1,img2,x,y):
    if insertionPossible(img1,img2,x,y)==True:
        (L,H)=img1.size
        (l,h)=img2.size
        for i in range (l):
            for j in range (h):
                (r,g,b)=Image.getpixel(img2,(i,j))
```

```
                if r!=255 and g!=255 and b!=255:
                    Image.putpixel(img1,(x+i,y+j),(r,g,b))
    else:
        print("Impossible")
```

Question 3.4

```
def insereTexteNet(img1,img2,x,y):
    if insertionPossible(img1,img2,x,y)==True:
        (L,H)=img1.size
        (l,h)=img2.size
        for i in range (l):
            for j in range (h):
                (r,g,b)=Image.getpixel(img2,(i,j))
                (R,G,B)=Image.getpixel(img1,(x+i,y+j))
                if r!=255 and g!=255 and b!=255:
                    Image.putpixel(img1,(x+i,y+j),((r*R)//255,(g*G)//255,(b*B)//255))
    else:
        print("Impossible")
```

TD VI
Boucle while

1 Lancer de dé

Question 1.1

```
def un_de():
    return int(random.uniform(1,7))

def nbLancers():
    cpt=1
    while (un_de()!=6):
        cpt=cpt+1
    return cpt
```

Question 1.2

```
def double():
    cpt=1
    while(un_de()!=un_de):
        cpt=cpt+1
    return cpt
```

Question 1.3

```
def moyenneTentativedouble(n):
    somme=0
    for x in range(n):
        tent=double()
        somme=somme+tent
    return (somme//n)
```

Question 1.4

```
def uniformite(n):
    L=[0,0,0,0,0,0,0]
    for y in range(n):
        x=un_de()
        L[x]=L[x]+1
    return L
```

2 Ecriture décimale

Question 2.1

```
def mystere(n):
    s=0
    while n>0:
        s=s+n%10
        n=n//10
    return s
```

Question 2.2

```
#Mystère 2705 renvoi 14 soit 2+7+0+5
```

Question 2.3

```
def nombreDeChiffres(n):
    cpt=0
    while(n!=0):
        n=n//10
        cpt=cpt+1
    return cpt
```

Question 2.4

```
def plusGrandChiffre(n):
    a=0
    while(n!=0):
        b=a
        a=n%10
        n=n//10
        if b>a:
            a=b
    return a
```

3 Racine carée

Question 3.1

```
def sqrt(n):
    a=1
    b=0
    cpt=0
    while b<n:
        a=a+2
        b=b+a
        cpt=cpt+1
    return cpt
```

4 Logarithme

Question 4.1

```
def logarithme_1(a,n):
    cpt=0
    if a==1:
        print ("Impossible, divison par 0")
        return False
    while a**cpt<n:
        cpt=cpt+1
    return cpt
def logarithme_2(a,n):
    return logarithme_1(a,n)-1
```

5 Pour les plus rapides

Question 5.1

```
def conway(n):
    c=[1]
    for j in range(n):
        L=[]
        i=0
        while i<len(c):
            cpt=0
            while(i+cpt<len(c) and c[i+cpt]==c[i]):
                cpt+=1
```

```
            L+=[cpt]
            L+=[c[i]]
            i+=cpt

        c=[]
        for i in L:
            c+=[i]
    return c
```

Question 5.2

```
def suiteComplexe(x,y,size,n):
    c=complex(x,y)*3/size-complex(2,1.5)
    z=c
    for i in range(n):
        z=z*z+c
    return z
```

Question 5.3

```
def suiteComplexeDiverge(x,y,size):
    n=0
    while n<256 and abs(suiteComplexe(x,y,size,n))<2:
        n=n+1
    return n
```

Question 5.4

```
def mandelbrot(size):
    nimg=new("RGB",(size,size))
    avt=0
    for x in range(size):
        for y in range(size):
            g=suiteComplexeDiverge(x,y,size)
            Image.putpixel(nimg,(x,y),(g,g,g))
            avt=avt+1
        print(int(avt/(size*size)*100),"%")
    return nimg
```

Question 5.5

```
def mandelbrotCouleur(size):
    nimg=new("RGB",(size,size))
    avt=0
    for x in range(size):
        for y in range(size):
            n=suiteComplexeDiverge(x,y,size)
            Image.putpixel(nimg,(x,y),((n*8)%256,(n*32)%256,(n*64)%256))
            avt=avt+1
        print(int(avt/(size*size)*100),"%")
    return nimg
```

TD VII
Récursivité

1 Arbre

```
def arbre(L,n,a,b):
    down()
    forward(L)
    if n>1:
        left(a)
        arbre(L*b,n-1,a,b)
        right(2*a)
        arbre(L*b,n-1,a,b)
        left(a)
    up()
    speed("fastest")
    back(L)
```

2 Flocon de Koch

```
def flocon(L,n):
    if n==0:
        forward(L)
    else:
        flocon(L//3,n-1)
        left(60)
        flocon(L//3,n-1)
        right(120)
        flocon(L//3,n-1)
        left(60)
        flocon(L//3,n-1)
    speed("fastest")
```

3 Triangle de Sierpinski

```
def triangle(n,L):
    if n==0:
        for i in range(0,3):
            forward(L)
            left(120)
    if n>0:
        triangle(n-1,L/2)
        forward(L/2)
        triangle(n-1,L/2)
        backward(L/2)
        left(60)
        forward(L/2)
        right(60)
        triangle(n-1,L/2)
        left(60)
        backward(L/2)
        right(60)
    speed("fastest")
```

4 Drive

```
def logo_drive(L,n):
    down()
```

```
left(60)
if n==0:
    forward(L/2)
else:
    left(60)
    demi_hexa(L,n-1)
    right(60)
    demi_hexa(L,n-1)
    right(60)
    demi_hexa(L,n-1)
    demi_hexa(L,n-1)
speed("fastest")
```