Programme testant les 128 cas possibles de jeu de paramètres

```python
def norme1(G1,G2,a):
    A=sqrt(sousnorme(G1,G2,a))
    return(A)
def norme2(G1,G2,a,b):
    A=sqrt((sousnorme(G1,G2,a)+sousnorme(G1,G2,b))/2)
    return(A)
def norme3(G1,G2,a,b,c):
    A=sqrt((sousnorme(G1,G2,a)+sousnorme(G1,G2,b)+sousnorme(G1,G2,c))/3)
    return(A)
def norme4(G1,G2,a,b,c,d):
    A=sqrt((sousnorme(G1,G2,a)+sousnorme(G1,G2,b)+sousnorme(G1,G2,c)
+sousnorme(G1,G2,d))/4)
    return(A)
def norme5(G1,G2,a,b,c,d,e):
    A=sqrt((sousnorme(G1,G2,a)+sousnorme(G1,G2,b)+sousnorme(G1,G2,c)+sousnorme(G1,G2,d)
+sousnorme(G1,G2,e))/5)
    return(A)
def norme6(G1,G2,a,b,c,d,e,f):
    A=sqrt((sousnorme(G1,G2,a)+sousnorme(G1,G2,b)+sousnorme(G1,G2,c)+sousnorme(G1,G2,d)
+sousnorme(G1,G2,e)+sousnorme(G1,G2,f))/6)
    return(A)
def norme7(G1,G2):

A=sqrt((sousnorme(G1,G2,0)+sousnorme(G1,G2,1)+sousnorme(G1,G2,2)+sousnorme(G1,G2,3)+s
ousnorme(G1,G2,4)+sousnorme(G1,G2,5)+sousnorme(G1,G2,6))/7)
    return(A)

def creaNormes(n,a=0,b=0,c=0,d=0,e=0,f=0):
    if (n==1):
        for i in range(N-1):
            for j in range(N-1):
                Normes[i,j]=norme1(G[:,i],G[:,j],a)
                if i==j:
                    Normes[i,j]=0
    if (n==2):
        for i in range(N-1):
            for j in range(N-1):
                Normes[i,j]=norme2(G[:,i],G[:,j],a,b)
                if i==j:
                    Normes[i,j]=0
    if (n==3):
        for i in range(N-1):
            for j in range(N-1):
                Normes[i,j]=norme3(G[:,i],G[:,j],a,b,c)
                if i==j:
                    Normes[i,j]=0
    if (n==4):
        for i in range(N-1):
            for j in range(N-1):
                Normes[i,j]=norme4(G[:,i],G[:,j],a,b,c,d)
```

```python
                    if i==j:
                        Normes[i,j]=0
        if (n==5):
            for i in range(N-1):
                for j in range(N-1):
                    Normes[i,j]=norme5(G[:,i],G[:,j],a,b,c,d,e)
                    if i==j:
                        Normes[i,j]=0
        if (n==6):
            for i in range(N-1):
                for j in range(N-1):
                    Normes[i,j]=norme6(G[:,i],G[:,j],a,b,c,d,e,f)
                    if i==j:
                        Normes[i,j]=0
        if (n==7):
            for i in range(N-1):
                for j in range(N-1):
                    Normes[i,j]=norme7(G[:,i],G[:,j])
                    if i==j:
                        Normes[i,j]=0


creaNormes(7)
Normes7=np.copy(Normes)
Corr7=np.percentile(Normes7,10)



def calculR(Normes):
    R=0
    Corr=np.percentile(Normes,10)
    for i in range(N-1):
        for j in range(N-1):
            if Normes[i,j]<=Corr:
                if (i+1<N-1) and (j+1<N-1):
                    if Normes[i+1,j+1]>Corr:
                        R+=1
    return(R)

def calculR7(Normes):
    R=0
    Corr=np.percentile(Normes,10)
    for i in range(N-1):
        for j in range(N-1):
            if Normes[i,j]<=Corr:
                if Normes7[i,j]>Corr7:
                    R+=1
    return(R)



L=[]

for i in range(7):
```

```python
        creaNormes(1,i)
        L.append([calculR(Normes),calculR7(Normes),i])

for i in range(7):
    for j in range(0,i):
        creaNormes(2,i,j)
        L.append([calculR(Normes),calculR7(Normes),[i,j]])

for i in range(7):
    for j in range(0,i):
        for k in range(0,j):
            creaNormes(3,i,j,k)
            L.append([calculR(Normes),calculR7(Normes),[i,j,k]])

for i in range(7):
    for j in range(0,i):
        for k in range(0,j):
            for l in range(0,k):
                creaNormes(4,i,j,k,l)
                L.append([calculR(Normes),calculR7(Normes),[i,j,k,l]])

for i in range(7):
    for j in range(0,i):
        for k in range(0,j):
            for l in range(0,k):
                for m2 in range(0,l):
                    creaNormes(5,i,j,k,l,m2)
                    L.append([calculR(Normes),calculR7(Normes),[i,j,k,l,m2]])

for i in range(7):
    for j in range(0,i):
        for k in range(0,j):
            for l in range(0,k):
                for m2 in range(0,l):
                    for n in range(0,m2):
                        creaNormes(6,i,j,k,l,m2,n)
                        L.append([calculR(Normes),calculR7(Normes),[i,j,k,l,m2,n]])

for i in range(7):
    for j in range(0,i):
        for k in range(0,j):
            for l in range(0,k):
                for m2 in range(0,l):
                    for n in range(0,m2):
                        for o in range(0,n):
                            creaNormes(7)
                            L.append([calculR(Normes),calculR7(Normes),[i,j,k,l,m2,n,o]])
```