

```

# TP2-corrigé.py

001 #Exercice n°1
002
003 acc=0
004 for i in range(1,100):
005     for j in range(1,100):
006         acc+=i/j
007 print(acc)
008
009 def commun(L1,L2):
010     L=[]
011     for i in L1:
012         for j in L2:
013             if i==j:
014                 L.append(i)
015     return(L)
016
017 #Exercice n°2
018
019 def ordrel(L):
020     n=len(L)
021     for j in range(1,n):
022         trier=True
023         for i in range(0,n-j):
024             if len(L[i])>len(L[i+1]):
025                 L[i],L[i+1]=L[i+1],L[i]
026                 trier=False
027         if trier:
028             break
029
030
031 def voyelle(chaine):
032     dico={'a':0,'e':0,'i':0,'o':0,'u':0,'y':0}
033     compt=0
034     for car in chaine:
035         if car in dico:
036             dico[car]=dico[car]+1
037     for car in dico:
038         compt=compt+dico[car]
039     return(compt)
040
041 def ordre2(L):
042     n=len(L)
043     for j in range(1,n):
044         trier=True
045         for i in range(0,n-j):
046             if voyelle(L[i])>voyelle(L[i+1]):
047                 L[i],L[i+1]=L[i+1],L[i]
048                 trier=False
049         if trier:
050             break
051
052 def ordre3(L):
053     n=len(L)
054     for j in range(1,n):
055         trier=True
056         for i in range(0,n-j):
057             k=0
058             while L[i][k]==L[i+1][k] and k<min(len(L[i])-1,len(L[i+1])-1):
059                 k=k+1
060             if L[i][k]>L[i+1][k]:
061                 L[i],L[i+1]=L[i+1],L[i]
062                 trier=False
063             if k==min(len(L[i])-1,len(L[i+1])-1):
064                 if len(L[i])>len(L[i+1]):
065                     L[i],L[i+1]=L[i+1],L[i]
066                 trier=False
067         if trier:
068             break
069
070 #Exercice n°3
071 clavier={'2':['a','b','c'],'3':['d','e','f'],'4':['g','h','i'],'5':['j','k','l'],'6':['m','n','o'],'7':
['p','q','r','s'],'8':['t','u','v'],'9':['w','x','y','z']}
072
073 def decodage(code):
074     for c1 in clavier[code[0]]:
075         for c2 in clavier[code[1]]:
076             for c3 in clavier[code[2]]:
077                 for c4 in clavier[code[3]]:
078                     for c5 in clavier[code[4]]:
079                         print(c1+c2+c3+c4+c5)
080
081
082 #Exercice n°4
083 def distance2(point):
084     x,y=point
085     return(x**2+y**2)
086
087 def compare(p1,p2):
088     d1=distance2(p1)
089     d2=distance2(p2)
090     if d1<d2:
091         return(-1)
092     elif d2<d1:
093         return(1)
094     else:
095         return(0)
096
097 def tri_points(L):
098     for j in range(1,len(L)):

```

```

099|         tri=True
100|         for i in range(0,len(L)-j):
101|             if compare(L[i],L[i+1])==1:
102|                 L[i],L[i+1]=L[i+1],L[i]
103|                 tri=False
104|         if tri:
105|             break
106|
107| #Exercice n°5
108| from random import randint
109| from time import time
110|
111| L1=[randint(1,100000) for i in range (100)]
112| L2=[randint(1,100000) for i in range (1000)]
113| L3=[randint(1,100000) for i in range (10000)]
114|
115| def tri_bulles(L):
116|     j=len(L)-1
117|     while j>0:
118|         tri=True
119|         for i in range(0,j):
120|             if L[i]>L[i+1]:
121|                 L[i],L[i+1]=L[i+1],L[i]
122|                 tri=False
123|         j=j-1
124|         if tri:
125|             j=0
126|
127| top=time()
128| #tri_bulles(L1)
129| L1.sort()
130| print(time()-top)
131|
132| top=time()
133| #tri_bulles(L2)
134| L2.sort()
135| print(time()-top)
136|
137| top=time()
138| #tri_bulles(L3)
139| L3.sort()
140| print(time()-top)
141|
142|
143|
144|
145|
146|
147|
148|
149|
150|

```