

Corrigés série révision DS N°2 2015-16

exercice1.py

```
from numpy import *

def Eratosthene (n):
    t=array(list(range(2,n+1)))
    L=[]
    for i in range(2,int(sqrt(n))+1):
        if(i==t[0]):
            L.append(t[0])
            t=array(list(filter(lambda x: x%i!=0,t)))
            #print("tab=",L)
            #print("i=",i,"t=",t)
    L+=list(t[:])
    return array(L)

#Appel
print(Eratosthene (int(input("n= "))))
```

exercice2.py

```
#import
from numpy import *

#Fonctions
def Syracuse (x=500):
    t=[]
    while x !=1:
        if(x%2==0):
            x//=2
        else:
            x=3*x+1
        t.append(x)
    return array(t)

#Appel
print(Syracuse (40))
print(type(Syracuse()))
```

exercice3.py

```
#import
from numpy import *

#Fonctions
g = lambda x: pow(x,4)-4*pow(x,3)-pow(x,2)+16*x-12

#Appel
res=list(map(g,list(range(-5,6))))
print(res)

#Autrement
L=[g(i) for i in range(-5,6)]
print(L)
```

exercice4.py

```
#import
from random import *
from numpy import *

#1
L=[]
for i in range(20):
    L.append(randint(-10,10))

#2
L2=list(filter(lambda x : x>=5 , L))

#3
T=array(list(map(lambda x : [x,L2.count(x)],set(L2))))

#Appel
print("L1 =",L)
print("L2 =",L2)
print("T =",T)
```

exercice5.py

```
#import
from numpy import *

s=(0,1,2,3,4,5,6,7,8,9)
c=2
def f(s,c):
    L=[]
    for i in range(0,len(s),c):
        L.append(list(s[i:i+c]))
    return array(L)
print(f(s,2))
```

exercice6.py

```
#import
#from math import *
import numpy as np
from math import *
#1
#print(format(math.pi, '.5g'))    # give 5 significant digits
#print(format(math.pi, '.2f'))    # give 2 digits after the point

epsilon = 0.0000001210
#2
def viete(epsilon):
    A = 1/sqrt(2)
    i=A
    P=A
    while 1:
        j=i
        i=sqrt(1/2+(1/2)*i)
        P=P *i
        if abs(i-j) <= epsilon :
            break
    return (2/P)

print(viete(epsilon))

#3
def euler(epsilon):
    k=0
    S=0
    while 1:
        S1=S
        k=k+1
        S=S+8/(1-16*(k**2))
        if abs(S-S1) <= epsilon :
            break
    return (4+S)

print(euler(epsilon))
```

exercice7.py

```
from numpy import *

def legendre(n):
    pass
```

exercice8.py

```
#import
from math import *

#1 : Saisie entier > 5
while 1 :
    try :
        m=int(input("n= "))
        if m>5:
            break
    except ValueError :
        print("Erreur : Donnee invalide")

#2
#Saisie de deux listes a et b contenant m entiers tous distinctes
#Definir une fonction saisie
def saisie(nomListe,m):
    print("Saisie de la liste "+nomListe)
    L=[]
    while m>0:
        try :
            n=int(input("Element "+str(m)+" = "))
            L.append(n)
        except ValueError :
            print("Erreur : Donnee invalide")
        else:
            m-=1
    return L

while 1:
    a=saisie("A",m)
    b=saisie("B",m)
    #print(a)
    #print(b)
    #print(a.intersection(b))
    if( len(set(a).intersection(set(b))) ==0):
        break
    else:
        print("Resaisie de deux listes !")

print("Liste A =",a)
print("Liste B =",b)

#3
f=lambda x: sum([ a[n] * cos(n* pi * x) + b[n]*sin(n*pi*x) for n in range(
    m)])

#4
s=[f(x) for x in range(-5,6)]
print("SequenceS =",s)
```

exercice9.py

```
#1
def valide(chaine):
    """Fonction qui retourne la validite d'une chaine saisie"""
    if (len(chaine)==0 or not chaine.isalpha()):
        return False
    for c in chaine:
        if (c.upper() not in "ATGC"):
            return False
    return True

#2
def saisie(msg):
    """Fonction qui saisie et retourne une chaine valide"""
    while 1:
        chaine = input("Saisie une "+msg+" ADN : ")
        if (valide(chaine)):
            break
    return chaine

#3
def proportion(ch,s):
    return ch.count(s)*100/len(ch)

#4
ch=saisie("chaine")
s=saisie("sequence")
print('Il y a ',proportion(ch,s),' % de "',s,'" dans votre chaine.', sep
      = "")
```