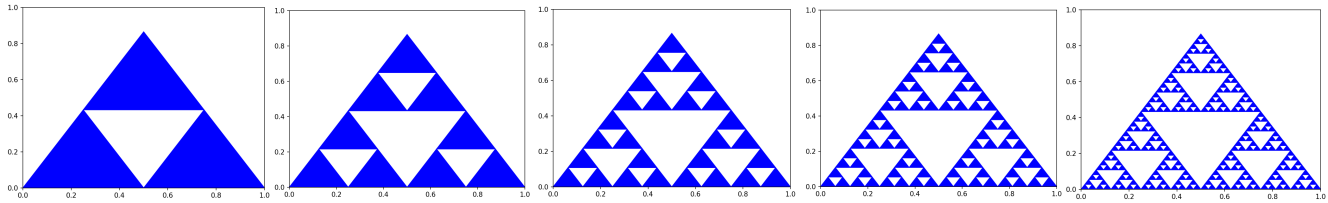


INFORMATIQUE - EXERCICES D'ENTRAÎNEMENT 2



Corrigés des...

2 - Exercices pour démarrer ... page 2

3 - Incontournables du programme - Sur les listes... page 3

- Recherche de minimum/maximum dans une liste
- Moyenne
- Nombre d'éléments égaux à...
- Algorithme de dichotomie (test d'appartenance)
- Algorithme de tri rapide (récursif)

4 - Incontournables du programme - Sur les suites... page 6

- Sommes des chiffres d'un entier
- Suite récurrente
- Suite de Fibonacci
- Coefficients binomiaux
- Algorithme de Syracuse
- Algorithme d'Euclide pour le calcul du PGCD

5 - Incontournables du programme - Sur les chaînes de caractères... page 8

- Nombre de voyelles dans une chaîne de caractères
- Palindrome ou non ?
- Recherche de sous-chaîne dans une chaîne

6 - Ultime incontournable - Dichotomie pour l'équation $f(x) = 0$... page 9

2 - DES EXERCICES POUR DÉMARRER

```
#####
```

```
# EXO 1
```

```
def F(n):
    if n == 0:
        return None
    else:
        print('Je dois ranger mon bureau')
        F(n-1)
```

```
#####
```

```
# EXO 2
```

```
def Somme(n):
    if n == 1:
        return 1
    else:
        return Somme(n-1) + (1/n)
```

```
# Test de validité (dans le shell): [Somme(k) for k in range(1,5)]
```

```
#####
```

```
# EXO 3
```

```
def Z2(n):
    if n == 1:
        return 1
    else:
        return Z2(n-1) + (1/n**2)
```

```
#####
```

```
# EXO 4
```

```
from random import randint
```

```
def SIM(n):
    alphabet = 'abcdefghijklmnopqrstuvwxyz'
    if n == 0:
        return None
    else:
        rgalea = randint(0,25)
        print(alphabet[rgalea], end='')
        SIM(n-1)
```

```
#####
```

3 - INCONTOURNABLES DU PROGRAMME - SUR LES LISTES

EXO 5

```
def MinR(L):
    if len(L) == 1:
        return L[0]
    else:
        if L[1] < L[0]:
            return MinR(L[1:len(L)])
        else:
            return MinR([L[0]]+L[2:len(L)])
```

#####

EXO 6

```
def MaxR(L):
    if len(L) == 1:
        return L[0]
    else:
        if L[1] > L[0]:
            return MaxR(L[1:len(L)])
        else:
            return MaxR([L[0]]+L[2:len(L)])
```

#####

EXO 7

```
def SomR(L):
    if len(L) == 0:
        return 0
    else:
        return L[0] + SomR(L[1:len(L)])
```

```
def MoyR(L):
    return SomR(L)/len(L)
```

#####

EXO 8

```
def NB4R(L):
    if L == []:
        return 0
    else:
        if L[0] == 4:
            return 1 + NB4R(L[1:])
        else:
            return NB4R(L[1:])
```

EXO 9

```
def NBR(L,n):
    if L == []:
        return 0
    else:
        if L[0]==n:
            return 1 + NBR(L[1:],n)
        else:
            return NBR(L[1:],n)
```

#####

EXO 10

```
def PairsR(L):
    if L == []:
        return 0
    else:
        if L[0]%2 == 0:
            return 1 + PairsR(L[1:])
        else:
            return PairsR(L[1:])
```

#####

EXO 11

```
def DiR(x,L):
    if L == []:
        return False
    p = len(L)//2
    if x == L[p]:
        return True
    elif x < L[p]:
        return DiR(x,L[:p])
    else:
        return DiR(x,L[p+1:])
```

```
#####
```

```
# EXO 12
```

```
def separe(L):  
    Lg = [L[k] for k in range(1, len(L)) if L[k] <= L[0]]  
    Ld = [L[k] for k in range(1, len(L)) if L[k] > L[0]]  
    return (Lg, Ld)
```

```
#####
```

```
# EXO 13
```

```
def TRAP(L):  
    if len(L) == 1 or len(L) == 0:  
        return L  
    else:  
        return TRAP(separe(L)[0]) + [L[0]] + TRAP(separe(L)[1])
```

```
#####
```

4 - INCONTOURNABLES DU PROGRAMME - SUR LES SUITES

EXO 14

```
def SCHIFFR(n):  
    if n < 10:  
        return n  
    else:  
        return (n%10) + SCHIFFR(n//10)
```

#####

EXO 15

```
def F12(n):  
    if n == 0:  
        return 2  
    else:  
        return (2 * F12(n-1) + 1) / 3
```

#####

EXO 16

```
def F13(n):  
    if n == 0:  
        return 1  
    elif n == 1:  
        return 3  
    else:  
        return F13(n-1) - F13(n-2) + 1
```

#####

EXO 17

```
def FiboR(n):  
    if n in [0, 1]:  
        return n  
    else:  
        return FiboR(n-1) + FiboR(n-2)
```

EXO 18

```
def BINR(n,p):
    if p==0:
        return 1
    elif n==0:
        return 0
    else:
        return BINR(n-1,p) + BINR(n-1,p-1)
```

#####

EXO 19

```
def SYR(n,x):
    if n ==0:
        return x
    else:
        if x%2==0:
            return SYR(n-1,x//2)
        else:
            return SYR(n-1,3*x+1)
```

#####

EXO 20

```
def PGCD(a,b):
    if a<b:
        a,b = b,a
    if b ==0:
        return a
    else:
        return PGCD(b,a%b)
```

5 - INCONTOURNABLES DU PROGRAMME - SUR LES CHAÎNES DE CARACTÈRES

EXO 21

```
def NBVOYR(mot):
    if mot == "":
        return 0
    else:
        if mot[0] in 'aeiouyAEIOUY':
            return 1 + NBVOYR(mot[1:])
        else:
            return NBVOYR(mot[1:])
```

#####

EXO 22

```
def PAL(mot):
    if len(mot) in [0,1]:
        return True
    else:
        if mot[0] == mot[-1]:
            return PAL(mot[1:len(mot)-1])
        else:
            return False
```

#####

EXO 23

```
def SCHR(mot1,mot2):
    if len(mot1) < len(mot2):
        return False
    else:
        if mot2 == mot1[:len(mot2)]:
            return True
        else:
            return SCHR(mot1[1:],mot2)
```


6 - ULTIME INCONTOURNABLE - DICHOTOMIE POUR L'ÉQUATION $f(x) = 0$

EXO 24

```
from math import *
```

```
def f(x):
    return x**2-2
```

```
def DICHOR(f,a,b,epsi):
    if f(a)*f(b) > 0:
        return 'Wazaaaaaa'
    else:
        if abs(b-a) < epsi:
            return (a,b)
        else:
            m =(a+b)/2
            if f(a)*f(m)<=0:
                return DICHOR(f,a,m,epsi)
            else:
                return DICHOR(f,m,b,epsi)
```

#####

EXO 25 — Pour conclure en beauté!

Mais que fait cette mystérieuse fonction récursive ?...

```
def LRB(n):
    if n == 0:
        return None
    else:
        print("C'est dans ",n," ans, je m'en irai , j'entends le loup et le renard chanter")
        LRB(n-1)
```