

Calculer dans le shell et en boucle

Exercice 1

Dans le shell de python, évaluer l'expression

$$x^3 - x^2 + x - 1$$

pour $x = 0.457$.

Exercice 2

Dans le shell de python, évaluer l'expression

$$1/x - x(1 - 1/x)$$

pour $x = 0.333$.

Exercice 3

Dans le shell de python, évaluer l'expression

$$2 \Pi^4 + 3 \Pi^2 - 2 \Pi^3 - 2 - \Pi \cdot (-\Pi)$$

pour $\Pi = 0.3572189$.

La lettre Π est la majuscule de π . On pourra utiliser la variable `Pi` dans le shell pour désigner cette lettre.

Exercice 4

Dans le shell de python, évaluer l'expression

$$\Phi^3 - \frac{1}{\Phi} + 5(\Phi^2 - 2)$$

pour $\Phi = 31/17$.

La lettre Φ est la majuscule de ϕ , la lettre phi. On pourra utiliser la variable `Phi` dans le shell pour désigner cette lettre.

On pourra utiliser le module `fractions` de python pour obtenir le résultat sous forme de fraction.

Exercice 5

Dans le shell de python, évaluer l'expression

$$4\pi \varepsilon \frac{1}{\frac{1}{R_1} - \frac{1}{R_2}}$$

pour

$$\varepsilon = 8.85418782 \cdot 10^{-12} \quad R_1 = 0.053 \quad R_2 = 0.0012$$

La lettre ε est la lettre epsilon. On pourra utiliser la variable `epsi` dans le shell pour désigner cette lettre.

Exercice 6

Dans le shell, affecter à la variable `i` la valeur 0. Quelle est la valeur de `i` après l'évaluation de l'expression ci-dessous ?

```
i = i + 1
```

Que se passe-t-il à chaque nouvelle évaluation de cette expression ?

Exercice 7

Dans le shell, affecter à la variable `i` la valeur 0. Affecter ensuite la valeur 10 à la variable `n`. On écrira ensuite la boucle ci-dessous.

```
while i < n:  
    i = i + 1
```

Que se passe-t-il au moment de l'évaluation de ces deux lignes ? Quelle est la valeur de `i` après l'exécution de ces lignes de code ?

Exercice 8

Dans le shell, affecter la valeur 10 à la variable `n`. On écrira ensuite la boucle ci-dessous.

```
for i in range(n):  
    pass
```

Que se passe-t-il au moment de l'évaluation de ces deux lignes ? Quelle est la valeur de `i` après l'exécution de ces lignes de code ?

Faire évaluer l'expression suivante: `list(range(10))`

Exercice 9

Dans un fichier, écrire le code ci-dessous :

```
n = 15  
i = 0  
while i < n:  
    print(i)  
    i = i + 1  
for i in range(n):  
    print(i)
```

Observer ce qui s'affiche à la console. Faire varier la valeur de `n`.

Exercice 10

Dans le shell, affecter à la variable `x` la valeur $\pi/4$. À l'aide d'une boucle, faire évaluer 100 fois l'expression `x = cos(x)`.

Que peut-on observer ?

Exercice 11

Soit $v = 1.4$. Vu que

$$1.4^2 = 1.96$$

on peut dire que v est une approximation de la racine de 2. On peut montrer que le nombre donné par

$$\frac{1}{2} \left(\frac{2}{v} + v \right)$$

est une « meilleure » approximation de la racine de 2. Dans le shell de python, calculer ce nombre.

En itérant ce processus dans le shell, trouver l'approximation du nombre $\sqrt{2}$ qui correspond le mieux à celle donnée par le shell python en évaluant l'expression

`2**(1/2)`

Exercice 12

On considère la liste de nombres donnée par

$$1 + \frac{1}{1} \quad 1 + \frac{1}{1 + \frac{1}{1}} \quad 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1}}} \quad 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1}}}} \quad \dots$$

Calculer dans le shell python une valeur approchée de chacun de ces nombres.

Exercice 13

Trouver la fraction continue qui donne le cinquième nombre de la liste de l'exercice précédent. Calculer une valeur approchée de ce nombre.

Exercice 14

Dans le shell python, calculer la valeur de $(1 + \sqrt{5})/2$. Comparer avec la valeur trouvée à l'exercice précédent.

Exercice 15

Écrire un programme en python qui permet de trouver le code décimal de la $n^{\text{ème}}$ valeur de la suite de l'exercice 12.

Exercice 16

Quel terme de la suite de l'exercice 12 faut-il choisir pour obtenir un nombre dont toutes les décimales sont égales à celles du nombre affiché dans le shell de python pour le nombre $(1 + \sqrt{5})/2$?

Exercice 17

On considère la liste de nombres donnée par

$$1 + \frac{1}{2} \quad 1 + \frac{1}{2 + \frac{1}{2}} \quad 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2}}} \quad 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2}}}} \quad \dots$$

Calculer dans le shell python une valeur approchée de chacun de ces nombres.

Exercice 18

Trouver la fraction continue qui donne le cinquième nombre de la liste de l'exercice précédent. Calculer une valeur approchée de ce nombre.

Exercice 19

Dans le shell python, calculer la valeur de la racine de 2. Comparer avec la valeur trouvée à l'exercice précédent.

Exercice 20

Écrire un programme en python qui permet de trouver le code décimal de la $n^{\text{ème}}$ valeur de la suite de l'exercice 17.

Exercice 21

Quel terme de la suite de l'exercice 17 faut-il choisir pour obtenir un nombre dont toutes les décimales sont égales à celles du nombre affiché dans le shell de python pour la racine de 2 ?

Exercice 22

On considère la suite d'expressions ci-dessous :

$$1 + 1/(1 + 1/2)$$

$$1 + 1/(1 + 1/(2 + 1/1))$$

$$1 + 1/(1 + 1/(2 + 1/(1 + 1/2)))$$

$$1 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/1))))$$

$$1 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/(1 + 1/2)))))$$

$$1 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/1)))))$$

$$1 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/(1 + 1/2)))))$$

En écrivant un programme en python, se convaincre du fait que cette suite de nombres tend vers $\sqrt{3}$.

Donner les fractions continues correspondantes.

Exercice 23

On considère les quatre nombres

$$3 + \frac{1}{7} \quad 3 + \frac{1}{7 + \frac{1}{15}} \quad 3 + \frac{1}{7 + \frac{1}{15 + \frac{1}{1}}} \quad 3 + \frac{1}{7 + \frac{1}{15 + \frac{1}{1 + \frac{1}{292}}}}$$

Calculer dans le shell python une valeur approchée de chacun de ces nombres.

On donne la liste

$$[3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, 2, 1, 1, 2, 2, 2, 2, 1, 84, 2, 1, 1, 15]$$

En construisant une fraction continue à l'aide de cette liste, trouver une approximation de π . Cette approximation est-elle meilleure que celle donnée par le module `math` de python ?

Exercice 24

Faire calculer par la machine les sommes suivantes :

$$1$$

$$1 + 2$$

$$1 + 2 + 3$$

$$1 + 2 + 3 + 4$$

$$1 + 2 + 3 + 4 + 5$$

$$1 + 2 + 3 + 4 + 5 + 6$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$$

On va chercher à automatiser les calculs ci-dessus et leur affichage.

Exercice 25

Écrire une boucle qui fait afficher les 50 premiers nombres entiers non nuls, avec un nombre par ligne.

Exercice 26

Écrire une boucle dans laquelle le compteur i varie entre 1 et 50. Cette boucle permet de stocker dans une variable s la somme des i premiers entiers. Faire afficher la somme au fur et à mesure du calcul.

Exercice 27

Écrire une boucle qui fait afficher la ligne ci-dessous.

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$$

Exercice 28

Écrire une boucle qui fait afficher les lignes ci-dessous.

$$1 = 1$$

$$1 + 2 = 3$$

$$1 + 2 + 3 = 6$$

$$1 + 2 + 3 + 4 = 10$$

$$1 + 2 + 3 + 4 + 5 = 15$$

$$1 + 2 + 3 + 4 + 5 + 6 = 21$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 = 66$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 = 78$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 = 91$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 = 105$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 = 120$$

Exercice 29

On donne a , b et c trois nombres réels avec $a \neq 0$. Écrire un *programme* python qui permet de calculer les deux solutions de l'équation

$$a x^2 + b x + c = 0$$

Exercice 30

On donne a , b et c trois nombres réels avec $a \neq 0$. En utilisant une *fonction* qui donne les deux solutions de l'équation

$$a x^2 + b x + c = 0$$

faire résoudre automatiquement les exercices 2.2.4 et 2.5.2 du cours d'algèbre 1MR.

Solutions

1

```
x = 0.457
x**3-x**2+x-1
-0.656405007
```

2

```
x = 0.333
1/x-x*(1-1/x)
3.6700030030030026
```

3

```
Pi = 0.3572189
2*Pi**4 + 3*Pi**2 - 2*Pi**3 - 2 - Pi*(-Pi)
-1.548178463229533
```

4

```
Phi = 31/17
Phi**3 - 1/Phi + 5*(Phi** - 2)
7.0189634711620075
import fractions
Phi = fractions.Fraction(31, 17)
Phi**3 - 1/Phi + 5*(Phi** - 2)
Fraction(33139285, 4721393)
33139285/4721393
7.018963471162007
```

5

```
from math import pi
epsi = 8.85418782*10**(-12)
R_1 = 0.053
R_2 = 0.0012
4*pi*epsi*1/(1/R_1-1/R_2)
-1.3661108800005497e-13
```

6 Après une itération, la valeur référencée par i est 1. À chaque nouvelle évaluation, la valeur de i augmente de 1.

7 La valeur de i est incrémentée 10 fois. Après exécution, la valeur référencée par i est 10.

8 La valeur de i prend toutes les valeurs entières comprises entre 1 et 9. Après exécution, la valeur référencée par i est 9.

9 –

10 –

11

```
v = 1.4
2**(1/2)
1.4142135623730951
v = 1/2*(2/v+v)
v
1.4142857142857141
v = 1/2*(2/v+v)
v
1.4142135642135643
v = 1/2*(2/v+v)
v
1.414213562373095
```

12

```
1 + 1/1
2
1 + 1/(1 + 1/1)
1.5
1 + 1/(1 + 1/(1 + 1/1))
1.6666666666666665
1 + 1/(1 + 1/(1 + 1/(1 + 1/1)))
1.6
1 + 1/(1 + 1/(1 + 1/(1 + 1/(1 + 1/1))))
1.625
1 + 1/(1 + 1/(1 + 1/(1 + 1/(1 + 1/(1 + 1/1))))))
1.6153846153846154
```

13

$$1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1}}}}}$$

$1 + 1/(1 + 1/(1 + 1/(1 + 1/(1 + 1/1)))) = 1.625$

Avec une boucle for :

```
x = 1
##Une boucle pour les 5 barres de fraction
for i in range(5):
    x = 1 + 1/x
print(x)
```

14

$$1.625 \simeq 1.618033988749895$$

L'approximation n'est pas très bonne: 1 décimale commune.

15 Avec une boucle:

```
n = 10
x = 1
for i in range(n):
    x = 1 + 1/x
```

On peut en faire une fonction :

```
def phi(n):
    x = 1
    for i in range(n):
        x = 1 + 1/x
    return x
```

16 C'est à partir du terme numéro 38 que l'on obtient un nombre dont les décimales correspondent à la valeur donnée par la machine pour le nombre $(1 + \sqrt{5})/2$.

```
from math import sqrt
r = (1+sqrt(5))/2
def phi(n):
    x = 1
    for i in range(n):
        x = 1 + 1/x
    return x

for i in range(1, 50):
    print(phi(i), i)
print(r)
```

17 1.5, 1.4, 1.4166666666666667, 1.4137931034482758

18

$$1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2}}}}}$$

$$1 + 1 / (2 + 1 / (2 + 1 / (2 + 1 / (2 + 1 / 2)))) = 1.4142857142857144$$

Avec une boucle for :

```
x = 2
##Une boucle pour les 4 premières barres de fraction
for i in range(4):
    x = 2 + 1/x
##La cinquième barre de fraction
x = 1 + 1/x
```

19

$1.4142857142857144 \simeq 1.4142135623730951$

Les quatre premières décimales sont identiques.

20 Avec une boucle :

```
n = 10
x = 2
for i in range(n - 1):
    x = 2 + 1/x
x = 1 + 1/x
```

On peut en faire une fonction :

```
def rac2(n):
    x = 2
    for i in range(n - 1):
        x = 2 + 1/x
    x = 1 + 1/x
    return x
```

21 C'est à partir du terme numéro 21 que l'on obtient un nombre dont les décimales correspondent à la valeur donnée par la machine pour la racine de 2.

```
from math import sqrt
r = sqrt(2)
def rac2(n):
    i = 0
    x = 2
    for i in range(n - 1):
        x = 2 + 1/x
    x = 1 + 1/x
    return x

for i in range(1, 30):
    print(rac2(i), i)
print(r)
```

22

```
##1ère ligne
x = 2
x = 1 + 1/x
x = 1 + 1/x

print(x)
print(1 + 1/(1 + 1/2))

##2ème ligne
x = 1
x = 2 + 1/x
x = 1 + 1/x
x = 1 + 1/x

print(x)
print(1 + 1/(1 + 1/(2 + 1/1)))
##3ème ligne
x = 2
x = 1 + 1/x
x = 2 + 1/x
x = 1 + 1/x
x = 1 + 1/x

print(x)
print(1 + 1/(1 + 1/(2 + 1/(1 + 1/2))))

##4ème ligne
x = 1
x = 2 + 1/x
x = 1 + 1/x
x = 2 + 1/x
x = 1 + 1/x
x = 1 + 1/x
print(x)

suite = [1, 2, 1, 2, 1, 1]
x = suite[0]
for n in suite[1:]:
    x = n + 1/x
print(x)

print(1 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/1)))))

##5ème ligne
x = 2
x = 1 + 1/x
```

```
x = 2 + 1/x
x = 1 + 1/x
x = 2 + 1/x
x = 1 + 1/x
x = 1 + 1/x
print(x)
```

```
x = 2
suite = [1, 2, 1, 2, 1, 1]
for n in suite:
    x = n + 1/x
print(x)
```

```
print(1 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/(1 + 1/2))))))
```

```
##6ème ligne
```

```
n = 6
x = 2 if n&1 else 1
suite = [1, 1]
for i in range(n - 1):
    suite = [2 - i%2] + suite
print(suite)
for k in suite:
    x = k + 1/x
print(x)
print(1 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/1))))))))
```

```
##7ème ligne
```

```
n = 7
x = 2 if n&1 else 1
suite = [1, 1]
for i in range(n - 1):
    suite = [2 - i%2] + suite
print(suite)
for k in suite:
    x = k + 1/x
print(x)
print(1 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/(1 + 1/(2 + 1/(1 + 1/2))))))))
```

```
##Ligne quelconque
```

```
n = 27
x = 2 if n&1 else 1
suite = [1, 1]
for i in range(n - 1):
    suite = [2 - i%2] + suite
print(suite)
for k in suite:
```

```

    x = k + 1/x
print(x)
print(3**(1/2))

```

23

```

3 + 1/7
3.142857142857143
3 + 1/(7 + 1/15)
3.141509433962264
3 + 1/(7 + 1/(15 + 1/1))
3.1415929203539825
3 + 1/(7 + 1/(15 + 1/(1 + 1/292)))
3.1415926530119025
3 + 1/(7 + 1/(15 + 1/(1 + 1/(292 + 1/1))))
3.141592653921421
3 + 1/(7 + 1/(15 + 1/(1 + 1/(292 + 1/(1 + 1/1))))))
3.1415926534674368

```

```

from math import pi
import decimal

```

```

denominateurs = [3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14,
                 2, 1, 1, 2, 2, 2, 2, 1, 84, 2, 1, 1, 15]
x = decimal.Decimal(denominateurs[-1])
n = len(denominateurs)
for i in range(n - 1):
    x = denominateurs[(n - 1) - (i + 1)] + 1/x
print(x)
print(pi)
print("3.1415926535897932384626433832795028841971693993751058209749445923")

```

24 La liste des sommes est : (1, 3, 6, 10, 21, 28, 36, 45, 55)

25

```

i = 1
n = 50

while i <= n:
    print(str(i))
    i = i + 1

```

26

```

i = 1
n = 50
s = 0
while i <= n:

```

```
s = s + i
print(s)
i = i + 1
```

27

```
i = 2
t = "1"
s = 1
n = 10
while i <= n:
    s = s + i
    t = t + " + " + str(i)
    i += 1
print(t)
```

28

```
i = 2
t = "1"
s = 1
n = 16
while i <= n:
    print(t, " = ", str(s), "\n")
    s = s + i
    t = t + " + " + str(i)
    i += 1
```

29

```
def solutions(a, b, c):
    delta = b**2 - 4*a*c
    x_1 = (-b + (delta)**(1/2))/(2*a)
    x_2 = (-b - (delta)**(1/2))/(2*a)
    return(x_1, x_2)
```

30

```
def sol_degre_2(a, b, c):
    delta = b**2 - 4*a*c
    return ((-b + delta**(1/2))/(2*a), (-b - delta**(1/2))/(2*a))
```

#Exercice 2.2.4

```
trinomes = [[1, 5, 6],
             [1, 5, 4],
             [1, -6, 8],
             [1, -2, -35],
             [9, 6, 1],
             [4, 5, 1],
```



```
[1, -2, -80],
[3, 7, 3],
[6, 5, 1],
[1, -22, 85],
[1, 1, 1],
[16, -72, 81],
[40, 3, -28],
[1, 9, -10],
[2, -5, -2],
[4, 25, -21]]
print("Exercice 2.2.4")
for t in trinomes:
    a, b, c = t
    x_1, x_2 = sol_degre_2(a, b, c)
    print(f"{str(a)}*(x - ({x_1})*(x - ({x_2}))")

#Exercice 2.5.2
trinomes = [[2, -7, -4],
            [6, 11, 4],
            [6, -25, -25],
            [6, -20, 25],
            [12, 23, -24],
            [5, 29/3, -14/3]]
print("Exercice 2.5.2")
for t in trinomes:
    a, b, c = t
    x_1, x_2 = sol_degre_2(a, b, c)
    print(f"{str(a)}*(x - ({x_1})*(x - ({x_2}))")
```