

## 1 Calcul de $\pi$

---

```
1 # -*- coding: utf-8 -*-
2
3 def calculPi(n):
4     """
5     y = calculPi(n)
6     calcul de pi à l'ordre n
7
8     >>> from math import fabs, pi
9     >>> fabs(pi - calculPi(1)) < 1.
10    True
11    >>> fabs(pi - calculPi(1000000)) < 1.e-6
12    True
13    """
14    assert type(n) is int and n >= 0
15
16    y = 2.
17    for k in range(1,n+1):
18        u = 4*k*k
19        y = y*u/(u-1)
20    return y
21
22 #-----
23 if __name__ == "__main__":
24     import doctest
25     doctest.testmod()
```

---

## 2 Conversion décimal $\rightarrow$ base $b$

---

```
1 # -*- coding: utf-8 -*-
2
3 def conversion(n,b=2,k=8):
4     """
5     code = conversion(n,b,k)
6     code en base b sur k bits de l'entier décimal n -> list
7
8     >>> conversion(23,2,8)
9     [0, 0, 0, 1, 0, 1, 1, 1]
10    >>> conversion(23,5,3)
11    [0, 4, 3]
12    >>> conversion(23,21,3)
13    [0, 1, 2]
14    >>> conversion(23,25,2)
15    [0, 23]
16    """
17    assert type(n) is int
18    assert type(b) is int
19    assert type(k) is int
20    assert n >= 0 and b > 1 and k > 0
21    assert n < b**k - 1
22
23    code = []
24    quotient = n
25    for i in range(k): code.append(0)
26
```

```
27     i = k - 1
28     while quotient != 0 and i >= 0:
29         code[i] = quotient%b
30         quotient = quotient//b
31         i = i - 1
32
33     return code
34
35 #-----
36 if __name__ == "__main__":
37     import doctest
38     doctest.testmod()
```

---

### 3 Quinconce

---

```
1 # -*- coding: utf-8 -*-
2
3 from turtle import *
4
5 #-----
6 def quinconce(n,m,r):
7     """
8     quinconce(n,m,r)
9     trace n rangées de m cercles de rayon r
10    disposés en quinconce
11    >>> quinconce(5,10,10)
12    """
13    assert type(n) is int and n > 0
14    assert type(m) is int and m > 0
15    assert type(r) is int and r > 0
16
17    for i in range(n) :
18        x0 = r*(i%2)
19        y0 = 2*i*r
20        for j in range(m) :
21            up()
22            goto(x0+2*j*r,y0)
23            down()
24            circle(r)
25    return
26
27 #-----
28 if __name__ == "__main__":
29     import doctest
30     doctest.testmod()
```

---

## 4 Coefficients de Kreweras

```
1. >>> for n in range(7):
        print(n,':',end=' ')
        for m in range(n+1):
            print(g(n,m),end=' ')
        print()
```

```
0 : 1
1 : 0 1
2 : 0 1 1
3 : 0 1 2 2
4 : 0 2 4 5 5
5 : 0 5 10 14 16 16
6 : 0 16 32 46 56 61 61
```

```
2. >>> 12*g(5,5)/g(6,6)
3.1475409836065573
```

## 5 Portée des variables

```
>>> x = 2
>>> print(x)
2
```

```
>>> y = f(x)
>>> print(x)
f 4
2
```

```
>>> z = g(x)
>>> print(x)
f 4
g 16
2
```

```
>>> t = h(x)
>>> print(x)
f 4
f 8
g 32
h 96
2
```

```
>>> x = 2
>>> print(x)
2
```

```
>>> x = f(x)
>>> print(x)
f 4
4
```

```
>>> x = g(x)
>>> print(x)
f 8
g 32
32
```

```
>>> x = h(x)
>>> print(x)
f 64
f 128
g 512
h 1536
1536
```

## 6 Exécution d'une fonction itérative

1. Il s'agit du tableau de Pascal des coefficients du binôme  $(x + y)^n$  pour les valeurs de  $n$  allant de 0 à 6.

2.  $c$  représente le  $p^{ième}$  coefficient du binôme  $(x + y)^n$  :

$$c = C_n^p = \binom{p}{n} = \frac{n!}{p!(n-p)!}.$$

```
>>> for n in range(7):  
      f(n)
```

```
1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1  
1 5 10 10 5 1  
1 6 15 20 15 6 1
```