Nom :	Prénom :	Groupe :

Durée : 90' Documents, calculettes, téléphones et ordinateurs interdits

1 Calcul de π

```python
# -*- coding: utf-8 -*-

def calculPi(n):
    """
    y = calculPi(n)
    calcul de pi à l'ordre n

    >>> from math import fabs, pi
    >>> fabs(pi - calculPi(1)) < 1.
    True
    >>> fabs(pi - calculPi(1000000)) < 1.e-6
    True
    """
    assert type(n) is int and n >= 0

    y = 2.
    for k in range(1,n+1):
        u = 4*k*k
        y = y*u/(u-1)
    return y

#--------------------------------------------------------------------------
if __name__ == "__main__":
    import doctest
    doctest.testmod()
```

2 Conversion décimal $\rightarrow$ base b

```python
# -*- coding: utf-8 -*-

def conversion(n,b=2,k=8):
    """
    code = conversion(n,b,k)
    code en base b sur k bits de l'entier décimal n -> list

    >>> conversion(23,2,8)
    [0, 0, 0, 1, 0, 1, 1, 1]
    >>> conversion(23,5,3)
    [0, 4, 3]
    >>> conversion(23,21,3)
    [0, 1, 2]
    >>> conversion(23,25,2)
    [0, 23]
    """
    assert type(n) is int
    assert type(b) is int
    assert type(k) is int
    assert n >= 0 and b > 1 and k > 0
```

```
    assert n < b**k - 1

    code = []
    quotient = n
    for i in range(k): code.append(0)

    i = k - 1
    while quotient != 0 and i >= 0:
        code[i] = quotient%b
        quotient = quotient//b
        i = i - 1

    return code

#-------------------------------------------------------------------------
if __name__ == "__main__":
    import doctest
    doctest.testmod()
```

3 Quinconce

```
# -*- coding: utf-8 -*-

from turtle import *

#-------------------------------------------------------------------------
def quinconce(n,m,r):
    """
    quinconce(n,m,r)
    trace n rangées de m cercles de rayon r
    disposés en quinconce
    >>> quinconce(5,10,10)
    """
    assert type(n) is int and n > 0
    assert type(m) is int and m > 0
    assert type(r) is int and r > 0

    for i in range(n) :
        x0 = r*(i%2)
        y0 = 2*i*r
        for j in range(m) :
            up()
            goto(x0+2*j*r,y0)
            down()
            circle(r)
    return

#-------------------------------------------------------------------------
if __name__ == "__main__":
    import doctest
    doctest.testmod()
```

4 Coefficients de Kreweras

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

```
1
0 1
0 1 1
0 1 2 2
0 2 4 5 5
0 5 10 14 16 16
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 $(x+y)^n$.

```
>>> 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
```