

NOM :	PRÉNOM :	GROUPE :
-------	----------	----------

DURÉE : 90'

DOCUMENTS, CALCULETTES, TÉLÉPHONES ET ORDINATEURS INTERDITS

## 1 Calcul de $\pi$

Définir une fonction qui calcule  $\pi$  à l'ordre  $n$  selon la formule :

$$\frac{\pi^2}{6} = 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} - \dots + \frac{1}{n^2} = \sum_{k=1}^n \frac{1}{k^2}$$

## 2 Conversion base $b \rightarrow$ décimal

Définir une fonction qui calcule la valeur décimale  $n$  d'un entier positif  $t$  codé en base  $b$ .

Exemples :

$b = 2$	$t = [0, 0, 0, 1, 0, 1, 1, 1]$	$\rightarrow n = 23$
$b = 5$	$t = [0, 0, 4, 3]$	$\rightarrow n = 23$
$b = 21$	$t = [1, 2]$	$\rightarrow n = 23$
$b = 25$	$t = [0, 0, 0, 0, 0, 23]$	$\rightarrow n = 23$

### 3 Polygones réguliers

Définir une fonction qui trace un polygone régulier à  $n$  côtés de longueur  $d$  à partir du point de coordonnées  $(x_0, y_0)$ . On utilisera les instructions de tracé à la *Logo*.



## 4 Courbes fractales

On considère la procédure `p` ci-contre :

1. On considère l'appel `p(1,300)` et le crayon initialement en  $(0,0)$  avec une direction de  $-90$  (vers le bas). Dessiner le résultat de cet appel.
2. On considère l'appel `p(3,300)` et le crayon initialement en  $(0,0)$  avec une direction de  $-90$  (vers le bas). Dessiner le résultat de cet appel.

```
def p(n,d):  
    assert type(n) is int  
    assert n >= 0  
    if n == 0: forward(d)  
    else:  
        p(n-1,d/3.)  
        right(60)  
        p(n-1,d/3.)  
        left(120)  
        p(n-1,d/3.)  
        right(60)  
        p(n-1,d/3.)  
    return
```



## 5 Portée des variables

On considère les fonctions **f**, **g** et **h** suivantes :

```
def f(x):  
    x = 3*x  
    print('f', x)  
    return x
```

```
def g(x):  
    x = 3*f(x)  
    print('g', x)  
    return x
```

```
def h(x):  
    x = 3*g(f(x))  
    print('h', x)  
    return x
```

Qu'affichent les appels suivants ?

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

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

`>>> z = g(x)`  
`>>> print(x)`

`>>> t = h(x)`  
`>>> print(x)`

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

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

`>>> x = g(x)`  
`>>> print(x)`

`>>> x = h(x)`  
`>>> print(x)`

## 6 Exécution d'une fonction itérative

On considère la procédure `f` définie ci-contre.

1. Qu'affiche l'instruction suivante?

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

2. Que représente `c` à la fin de chaque itération sur `p`?

```
#-----
def f(n):
#-----
    for p in range(n+1) :
        num = 1
        den = 1
        for i in range(1,p+1) :
            num = num*(n-i+1)
            den = den*i
        c = num/den
        print(c,end=' ') #--- affichage
    print()              #--- affichage
    return
#-----
```

`c`?

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