1. Premiers algorithmes

✯ EXERCICE 1

Suite arithmétique

Programme avec Python :

```
# suite arithmetique
a=input("raison")
u=input("premier terme")
n=input("N")

S=u
print 'terme de rang 0 = ',u

for i in range(1,n+1):
    u=a+u
    S=S+u
    print'terme de rang ',i,' = ',u
print'somme des ',n+1,' premiers termes = ',S
```

✯ EXERCICE 2

Suite géométrique

Programme avec Python :

```
# suite geometrique
a=input("raison")
u=input("premier terme")
n=input("N")

S=u
print 'terme de rang 0 = ',u

for i in range(1,n+1):
    u=a*u
    S=S+u
    print'terme de rang ',i,' = ',u
print'somme des ',n+1,' premiers termes = ',S
```

2. Exercice

✯ EXERCICE 3

On considère un triangle équilatéral ABC. Cette figure P_1 constitue l'étape 1.

Pour l'étape 2, on divise chaque côté en trois parties égales et on construit un nouveau triangle équilatéral en lieu et place du segment central. On obtient la figure P_2.

En itérant cette étape, on construit ainsi une suite de polygones $P_1, P_2, P_3, \ldots$

Programme avec Python :

```
def koch(l,n):
# découpe du segment
    if n<=0:
        forward(l)
    else:
        koch(l/3,n-1)
        left(60)
        koch(l/3,n-1)
        right(120)
        koch(l/3,n-1)
        left(60)
        koch(l/3,n-1)

def flocon(l,n):
# dessin du flocon
    koch(l,n)
    right(120)
    koch(l,n)
    right(120)
    koch(l,n)

# programme principal - récursif
from turtle import *
etape=input("numero etape du flocon de Von Koch")-1
taille1=100.
flocon(taille1,etape)
```

Pour le polygone P_n, on désigne par c_n le nombre de ses côtés, par l_n la longueur de chaque côté, par p_n son périmètre et par a_n son aire.
Les suites (p_n) et (a_n) admettent-elles une limite ?