

Corrigé du TP Informatique 06

Exercice 1

1. La variable `k` n'est pas modifiée. Le test `k<=n` est donc toujours vrai ce qui provoque une boucle infinie.

2. On saisit :

```
def f(n):  
    k,res=0,0  
    while k<=n:  
        res+=k  
        k+=1  
    return res
```

La fonction ainsi modifiée renvoie la valeur de la somme $\sum_{k=0}^n k$.

Exercice 2

1. La variable `res` est initialisée à 1 puis est multipliée à chaque passage dans la boucle par l'entier k , lui-même incrémenté de 1 à chaque passage. On en déduit que la fonction `g` réalise le calcul de $n!$.

2.(a) En échangeant les lignes 5 et 6 de la fonction, on incrémente d'abord k ce qui implique qu'on va effectuer $2 \times 3 \times \dots (n+1)$. La nouvelle version va donc renvoyer $(n+1)!$.

2.(b) Pour renvoyer $n!$, il suffit d'arrêter la boucle une itération plus tôt avec une inégalité stricte :

```
def g(n):  
    k,res=1,1  
    while k<n:  
        k+=1  
        res*=k  
    return res
```

Exercice 3

1. On saisit :

```
def somme1(n):  
    res=0  
    k=1  
    while k<=n:  
        res+=k**2  
        k+=2  
    return res
```

ou aussi

```
def somme1(n):  
    res=0  
    k=0  
    while 2*k+1<=n:  
        res+=(2*k+1)**2  
        k+=1  
    return res
```

2. On saisit :

```
def somme2s(n):  
    res=0  
    for k in range(1,n+1,2):  
        res+=k**2  
    return res
```

Exercice 4

1. On saisit :

```
def geom1(n,q):  
    res,qk=0,1  
    for k in range(n):  
        res+=qk  
        qk*=q  
    return res
```

2. On saisit :

```
def geom2(n,q):  
    res,k=0,0  
    qk=1  
    while k<n:  
        res+=qk  
        qk*=q  
        k+=1  
    return res
```

Exercice 5

On saisit :

```
def seuil(p):  
    k,s=0,0  
    while s<p:  
        k+=1  
        s+=k  
    return s,k
```

Exercice 6

1. On saisit :

```
def somme_log(n):  
    res,k=0,1  
    while k*np.log(k)<=n:  
        res+=k  
        k+=1  
    return res
```

2. Oui, c'est possible, mais on écrit une boucle **while** déguisée puisqu'on casse la boucle **for** :

```
def somme_log_bis(n):  
    res=0  
    for k in range(1,n):  
        if k*np.log(k)>n:  
            return res  
        res+=k
```

On est contraint de casser la boucle puisqu'on ne sait pas déterminer de réciproque à l'application $x \mapsto x \ln(x)$.

Exercice 7

1. On saisit

```
def val(p,n):  
    a,k=n,0  
    while a%p==0:  
        k+=1  
        a/=p  
    return k
```

2. Le cas le plus défavorable est celui où on rentre le plus de fois dans la boucle ce qui correspond à la situation où l'entier n est une puissance de p . Ainsi, pour $n = p^a$, on rentre a fois dans la boucle avec $a = \frac{\ln(n)}{\ln(p)}$ qu'on note $a = \log_p(n)$.