Boucles *for*, boucles *while*

Des boucles *while* et des sommes

1. Tester le code suivant et comprendre à quoi correspond l'affichage obtenu.

```
s = 0
n = 1
while n<=5:
    s = s+n
    n = n+1
print(s)
```

2. Modifier ce programme afin d'obtenir la somme des nombres de 1 à 100. Faire afficher le résultat et noter la valeur obtenue.

Réponse : _____

3. En s'inspirant de ce qui précède, écrire une fonction de la forme :

```
def somme(n):
    ...
    return s
```

qui calcule et renvoie la somme des nombres de 1 à n.

4. Noter le résultat obtenu en faisant afficher pour `somme(9)`. *Réponse :* _____

5. Toujours en s'inspirant de ce qui précède, écrire une fonction de la forme :

```
def somme_carres(n):
    ...
```

qui calcule et renvoie la somme des carrés nombres de 1 à n.

6. Noter le résultat obtenu pour **`somme_carres(9)`**. *Réponse :* _____

Des sommes de chiffres

7. Tester le code suivant et comprendre à quoi correspond l'affichage obtenu.

```
n = 147
while n>0:
    print(n%10)
    n = n//10
```

8. Modifier le code précédent pour qu'il calcule la somme des chiffres du nombre n. À l'aide de ce programme, déterminer la somme des chiffres du nombre 9998345667.

Réponse : _____

9. En s'inspirant de ce qui précède, écrire une fonction de la forme :

```
def somme_chiffres(n):
    ...
    return s
```

qui calcule et renvoie la somme des chiffres du nombre n.

10. Vérifier que **`somme_chiffres(1035)`** donne bien la valeur attendue.

11. Écrire sur le même principe une fonction **`somme_chiffres_cube(n)`** qui calcule et renvoie la somme des cubes des chiffres du nombre n.

12. Utiliser la fonction **`somme_chiffres_cube`** pour déterminer la somme des cubes des chiffres du nombre 2346.

Réponse : _____

13. À l'aide de la fonction précédente et d'une boucle *while*, déterminer le premier nombre $n \geqslant 2$ qui est égal à la somme des cubes de ses chiffres.

Réponse : _____

Boucles *for* et calcul des termes d'une suite

14. On considère la suite $(u_n)_{\geqslant 0}$ définie par

$$u_0 = 1$$
$$u_{n+1} = 2u_n + 1$$

Calculer (à la main) u_1, u_2 et u_3.

Réponse : _____

15. Tester le code suivant :

```
u = 1
print(u)
u = 2*u+1
print(u)
u = 2*u+1
```

Vérifier que les premiers termes de la suite sont affichés. Quel est le dernier terme affiché, est-ce u_1 ? u_2 ? u_3 ? À quel terme de la suite est maintenant égal `u` ?

Réponse : _____

16. Tester le code suivant :

```
u = 1
for k in range(0,2):
    print(u)
    u = 2*u+1
```

Vérifier que les premiers termes de la suite sont affichés. Quel est le dernier terme affiché, est-ce u_1 ? u_2 ? u_3 ?

Réponse : _____

17. Modifier le code de la manière suivante et le tester :

```
u = 1
for k in range(0,2):
    u = 2*u+1
print(u)
```

Le dernier terme affiché correspond à quel u_n ?

18. Modifier ce code pour qu'il affiche uniquement u_{15}. *Réponse :* _____

19. En s'inspirant de ce code, écrire une fonction `u(n)` qui calcule et renvoie la valeur de u_n. Vérifier en affichant `u(15)`.

Boucles imbriquées

20. Tester le code suivant et comprendre à quoi correspond l'affichage obtenu.

```
s = 0
for i in range(1,10):
    for j in range(1,10):
        s = s+i*j
```

21. Modifier le code précédent pour obtenir la somme de tous les produits $i \times j$ avec $1 \leqslant i \leqslant 10$ et $1 \leqslant j \leqslant i$.

Réponse : _____

Instruction conditionnelle *if* et somme des diviseurs

22. Tester le code suivant et commenter le résultat obtenu.

```
n = 145
for k in range(1,n+1):
    if n%k==0:
        print(k,"est un diviseur de",n)
```

23. En s'inspirant de ce code, écrire une fonction **`somme_diviseurs(n)`** qui calcule et renvoie la somme des diviseurs de n *avec* 1 *inclus mais* n *exclu.*

24. Avec la fonction **`somme_diviseurs`**, déterminer la somme des diviseurs de 262.

Réponse : _____

25. Déterminer les nombres entiers compris entre 1 et 100 qui sont égaux à la somme de leurs diviseurs.

Réponse : _____

26. Déterminer le plus petit nombre $n \geqslant 100$ qui est égal à la somme de ses diviseurs.

Réponse : _____

Années bissextiles

27. Une année n'est bissextile (comportant 366 jours) que dans l'un des deux cas suivants :

- si l'année est divisible par 4 et non divisible par 100 ;
- si l'année est divisible par 400.

Écrire une fonction **`bissextile(a)`** qui détermine et renvoie le nombre de jours de l'année **`a`**.

28. Utiliser cette fonction pour effectuer le total du nombre de jours entre les années 1990 et 2010 (comprises).

Réponse : _____

Corrections

Q 8.

```
n = 9998345667
s = 0
while n>0:
    s = s+n%10
    n = n//10
print(s)
```

```
66
```

Q 9.

```
def somme_chiffres(n):
    s = 0
    while n>0:
        s = s+n%10
        n = n//10
    return s
```

Q 10.

```
print(somme_chiffres(1035))
```

```
9
```

Q 11.

```
def somme_chiffres_cube(n):
    s = 0
    while n>0:
        s = s+(n%10)**3
        n = n//10
    return s
```

Q 12.

```
print(somme_chiffres_cube(2346))
```

```
315
```

Q 13.

```
n = 2
while n!=somme_chiffres_cube(n):
    n = n+1
print(n,somme_chiffres_cube(n))
```

```
153 153
```

Q 18.

```
u = 1
for k in range(0,15):
    u = 2*u+1
print(u)
```

```
65535
```

Q 19.

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

print(u(15))
```

```
65535
```

Q 21.

```
s = 0
for i in range(1,11):
    for j in range(1,i+1):
        s = s+i*j
print(s)
```

```
1705
```

Q 23.

```
def somme_diviseurs(n):
    s = 0
    for k in range(1,n):
        if n%k==0:
            s = s+k
    return s
```

Q 24.

```
print(somme_diviseurs(262))
```

```
134
```

Q 25.

```
for n in range(1,101):
    if n==somme_diviseurs(n):
        print(n)
```

```
6 28
```

Q 26.

```
n = 100
while n!=somme_diviseurs(n):
    n = n+1
print(n,somme_diviseurs(n))
```

```
496 496
```

Q 27.

```
def bissextile(a):
    if (a%4==0 and a%100!=0) or a%400==0:
        return 366
    else:
        return 365
```

Q 28.

```
s = 0
for a in range(1990,2011):
    s = s+bissextile(a)
print(s)
```

```
7670
```