TD N14 — Algorithmique

Première générale — Spécialité mathématiques

Objectif du TD. Lire, compléter, corriger et écrire des algorithmes liés aux suites, aux sommes, aux seuils et aux simulations. Utiliser les affectations, conditions, boucles bornées, boucles non bornées, fonctions Python et listes pour contrôler un calcul.

Méthode repère. Pour comprendre un programme, on fait une trace d'exécution : valeurs initiales, passage dans la boucle, mise à jour des variables, condition d'arrêt, valeur affichée.
Vigilances. En Python, `=` affecte une valeur, tandis que `==` teste une égalité. Une boucle `while` doit modifier une variable intervenant dans la condition, sinon elle risque de ne jamais s'arrêter.

A — Réactivation : lire et exécuter

1 ⋆ [REP]
Associer chaque instruction Python à son rôle.

`x = 4`	
`print(x)`	
`x == 4`	
`x = x + 1`	
`input()`	

Rôles possibles : afficher, affecter, tester une égalité, saisir, augmenter.

2 ⋆ [CAL, REP]
Faire tourner le programme à la main.

```
a = 5
b = 2
a = a + b
b = 3*a
print(a)
print(b)
```

Quelles valeurs sont affichées ?

3 ⋆ [CAL]
On considère le programme suivant.

```
u = 7
u = 2*u - 3
u = 2*u - 3
print(u)
```

Compléter :

$$u \leftarrow 7 \longrightarrow \longrightarrow$$

Quelle valeur est affichée ?

4 ⋆ [REP]
Dire si chaque boucle est **bornée** ou **non bornée**.

```
for i in range(5):
    print(i)
```

```
while u < 100:
    u = 2*u
```

5 ⋆ [CAL, REP]
Compléter la trace d'exécution.

```
u = 3
for i in range(4):
    u = u + 5
print(u)
```

étape	valeur de u
départ	3
après 1 passage	
après 2 passages	
après 3 passages	
après 4 passages	

6 ⋆ [CAL, REP]
Que renvoie la fonction suivante ?

```
def f(x):
    return 3*x + 2
```

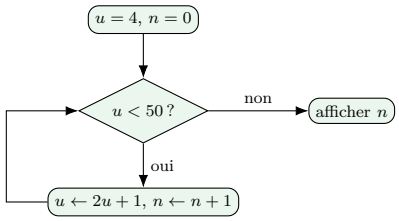
Calculer mentalement :

$$f(0), \qquad f(4), \qquad f(-1).$$

7 ⋆ [COM]
Expliquer en une phrase la différence entre :

`n = n + 1` et `n == n + 1`.

8 ⋆ [REP, CAL]
On donne le schéma de boucle suivant.



Calculer les valeurs successives de u, puis la valeur affichée.

B — Boucles, suites et seuils

9 ⋆ [CAL, REP]
Le programme suivant calcule des termes d'une suite.

```
u = 2
for n in range(5):
    u = u + 3
print(u)
```

1. Quelle suite est définie par ce programme ?
2. Quelle valeur est affichée ?
3. Cette valeur correspond-elle à u_4 ou à u_5 si $u_0 = 2$?

10 ⋆ [CAL, REP]
Même travail avec le programme suivant.

```
u = 5
for n in range(4):
    u = 2*u
print(u)
```

11 ⋆⋆ [REP, CAL]
Compléter le programme pour afficher les dix premiers termes de la suite définie par :

$$u_0 = 7, \qquad u_{n+1} = u_n + 4.$$

```
u = ...
for n in range(...):
    print(...)
    u = ...
```

12 ⋆⋆ [REP, CAL]
Compléter le programme pour afficher $u_0, u_1, \dots, u_8$, où :

$$u_0 = 3, \qquad u_{n+1} = 1.5u_n.$$

```
u = ...
for n in range(...):
    print(n, u)
    u = ...
```

13 ⋆⋆ [REP, CAL]
On veut déterminer le premier rang n tel que $u_n \geqslant 100$, pour :

$$u_0 = 4, \qquad u_{n+1} = u_n + 7.$$

Compléter le programme.

```
u = ...
n = ...
while ...:
    u = ...
    n = ...
print(n)
```

14 ⋆⋆ [REP, CAL]
On veut déterminer le premier rang n tel que $u_n \geqslant 1000$, pour :

$$u_0 = 12, \qquad u_{n+1} = 1.25u_n.$$

Compléter.

```
u = ...
n = ...
while ...:
    u = ...
    n = ...
print(n, u)
```

15 ⋆⋆ [CAL, REP]
Faire une trace d'exécution du programme.

```
u = 10
n = 0
while u < 50:
    u = 1.4*u
    n = n + 1
print(n)
```

n	u
0	10
1	
2	
3	
4	
5	

Que représente la valeur affichée ?

16 ⋆⋆ [CAL, REP]
On considère le programme suivant.

```
S = 0
for k in range(1, 11):
    S = S + k
print(S)
```

1. Quelle somme est calculée ?
2. Quelle valeur est affichée ?
3. Modifier mentalement le programme pour calculer $1 + 2 + \cdots + 100$.

17 ⋆⋆ [REP, CAL]
Compléter le programme pour calculer :

$$3 + 6 + 12 + 24 + \cdots + 3 \times 2^{10}.$$

```
u = ...
S = ...
for n in range(...):
    u = ...
    S = ...
print(S)
```

18 ⋆⋆ [RAI, COM]
Un élève propose le programme suivant pour trouver le premier rang tel que $u_n \geqslant 100$, avec $u_0 = 5$ et $u_{n+1} = u_n + 6$.

```
u = 5
n = 0
while u < 100:
    n = n + 1
print(n)
```

1. Pourquoi ce programme ne convient-il pas ?
2. Quelle ligne faut-il ajouter ?
3. Où faut-il l'ajouter ?

C — Fonctions Python, listes et simulations

19 ⋆⋆ [REP, CAL]
On considère la fonction suivante.

```
def terme_arith(u0, r, n):
    u = u0
    for k in range(n):
        u = u + r
    return u
```

1. Calculer à la main `terme_arith(4,3,5)`.
2. Quelle formule mathématique cette fonction utilise-t-elle implicitement ?

20 ⋆⋆ [REP, CAL]
Écrire une fonction `terme_geom(u0,q,n)` qui renvoie le terme u_n de la suite géométrique définie par :

$$u_0 = u0, \qquad u_{n+1} = qu_n.$$

21 ⋆⋆ [REP, CAL]
On considère la fonction suivante.

```
def seuil(u0, q, A):
    u = u0
    n = 0
    while u < A:
        u = q*u
        n = n + 1
    return n
```

1. Que renvoie `seuil(10,2,100)` ?
2. Interpréter le résultat en termes de suite.
3. Que se passe-t-il si $q = 1$ et $u0 < A$?

22 ⋆⋆ [RAI, COM]
Un élève écrit :

```python
def seuil(u0, r, A):
    u = u0
    n = 0
    while u <= A:
        u = u + r
    return n
```

Repérer deux erreurs et proposer une version corrigée.

23 ⋆⋆ [REP, CAL]
On considère le programme suivant.

```python
L = []
u = 3
for n in range(6):
    L.append(u)
    u = 2*u + 1
print(L)
```

1. Compléter la liste affichée.
2. Pourquoi l'instruction `L.append(u)` est-elle placée avant la mise à jour de u ?

24 ⋆⋆ [REP, CAL]
Écrire une fonction `liste_suite(u0,r,N)` qui renvoie la liste :

$$[u_0, u_1, \dots, u_N]$$

pour la suite arithmétique de premier terme u_0 et de raison r.

25 ⋆⋆⋆ [MOD, REP]
Un capital de 500 euros augmente de 4 % par an. On veut écrire une fonction qui renvoie le premier nombre d'années nécessaire pour dépasser un montant A.

1. Identifier la suite utilisée.
2. Compléter la fonction.

```python
def attente(A):
    C = ...
    n = ...
    while ...:
        C = ...
        n = ...
    return n
```

26 ⋆⋆⋆ [REP, CAL]
On veut simuler un lancer de dé équilibré à six faces.

```python
import random

def de():
    return random.randint(1, 6)
```

1. Que peut renvoyer la fonction `de()` ?
2. Écrire un programme qui lance le dé 20 fois et stocke les résultats dans une liste.
3. Quelle instruction permet de compter le nombre de 6 obtenus dans la liste L ?

27 ⋆⋆⋆ [MOD, REP]
On simule un jeu : on gagne 5 euros si le dé donne 6, sinon on perd 1 euro.

1. Compléter la fonction `gain()`.
2. Écrire une phrase décrivant la variable simulée.

```python
import random

def gain():
    d = random.randint(1, 6)
    if ...:
        return ...
    else:
        return ...
```

28 ⋆⋆⋆ [CH, REP]
On veut estimer le gain moyen du jeu précédent sur 1000 parties. Compléter.

```python
S = 0
for i in range(...):
    S = S + ...
m = S / ...
print(m)
```

Expliquer pourquoi le résultat peut changer lorsqu'on relance le programme.

D — Problèmes et synthèse

29 ⋆⋆⋆ [MOD, CAL, REP]
Une espèce invasive couvre 12 m^2 d'un lac au départ. Chaque semaine, la surface couverte augmente de 35 %. On veut savoir au bout de combien de semaines elle dépassera 500 m^2.

1. Écrire la relation de récurrence vérifiée par la suite.
2. Compléter l'algorithme en langage naturel.
3. Écrire ensuite un programme Python.

$u \leftarrow 12$
$n \leftarrow 0$
Tant que .....
  $u \leftarrow$ .....
  $n \leftarrow$ .....
Fin Tant que
Afficher .....

30 ⋆⋆⋆ [MOD, RAI]
Une association reçoit 200 euros de dons le premier mois. Chaque mois, les dons augmentent de 15 euros. Elle veut savoir au bout de combien de mois le total cumulé dépassera 3000 euros.

1. Faut-il comparer un terme ou une somme à 3000 ?
2. Compléter le programme.

```python
u = 200
S = ...
n = 0
while ...:
    n = ...
    u = ...
    S = ...
print(n, S)
```

31 ⋆⋆⋆ [CH, RAI, COM]
Un élève écrit le programme suivant pour calculer la somme des termes $u_0 + \cdots + u_{10}$, avec $u_0 = 4$ et $u_{n+1} = 3u_n$.

```python
u = 4
S = 0
for n in range(10):
    u = 3*u
    S = S + u
print(S)
```

1. Quelle somme ce programme calcule-t-il réellement ?
2. Pourquoi u_0 n'est-il pas pris en compte ?
3. Proposer une correction.

32 ⋆⋆⋆ [MOD, REP]
On étudie la suite définie par :

$$u_0 = 100, \qquad u_{n+1} = 0{,}85u_n + 12.$$

1. Cette suite est-elle arithmétique ? géométrique ? Justifier.
2. Écrire une fonction `terme(n)` qui renvoie u_n.
3. Écrire une fonction `liste(N)` qui renvoie la liste des termes de u_0 à u_N.

4. À quoi peut servir une liste de termes dans l'étude d'une suite ?

33 ⋆⋆⋆ [CH, MOD, REP]

On veut conjecturer le comportement de la suite :

$$u_0 = 2, \qquad u_{n+1} = \frac{1}{2}u_n + 3.$$

1. Compléter le programme pour afficher les 20 premiers termes.
2. À l'aide des valeurs affichées, formuler une conjecture.

```
u = ...
for n in range(...):
    print(n, u)
    u = ...
```

34 ⋆⋆⋆ [CH, MOD, RAI]

Une urne contient 3 boules rouges et 7 boules bleues. On tire une boule au hasard. Si elle est rouge, on gagne 4 euros ; sinon on perd 2 euros.

1. Compléter la fonction `gain()` qui simule un tirage.
2. Compléter ensuite le programme qui simule 500 tirages et affiche le gain moyen.
3. Que devrait-on calculer en probabilités pour contrôler ce résultat ?

```
import random

def gain():
    x = random.random()
    if x < ...:
        return ...
    else:
        return ...

S = 0
for i in range(...):
    S = ...
print(...)
```

35 ⋆⋆⋆ [CH, COM]

Bilan de méthode. Rédiger en cinq lignes maximum une fiche expliquant comment lire un programme, faire une trace, puis interpréter la valeur affichée. Faire apparaître les mots : *affectation*, *boucle*, *condition*, *seuil*, *variable*.