

Algorithmique (2ème partie)

I

Boucle bornée

Définition I.1

Une **boucle bornée** (aussi appelée boucle « pour ») est un type de boucle qui permet de répéter un bloc d'instructions un certain nombre de fois défini à l'avance.

Une boucle bornée s'écrit avec le mot-clé « Pour » en langage naturel et avec le mot-clé « for » en Python :

Langage naturel	Python
Pour K allant de x à y Instructions Fin Pour	for k in range(x,y+1): Instructions ...

Exemple I.1 — On donne l'algorithme suivant :

```

Pour K allant de 0 à 10
| A ← 2 * K
| Afficher A
Fin Pour
  
```

1. Traduire cet algorithme en Python.

2. Que va afficher cet algorithme ? Quel est son rôle ?

→ À rédiger

Exemple I.2 — Ecrire en langage naturel puis en Python un algorithme qui permet de calculer la somme de tous les nombres entiers de 1 à 20.

→ À rédiger

II

Boucle non bornée

Définition II.1

Une **boucle non bornée** (aussi appelée boucle « tant que ») est un type de boucle qui permet de répéter un bloc d'instructions tant qu'une certaine condition est vérifiée.

Une boucle non bornée s'écrit avec le mot-clé « Tant que » en langage naturel et avec le mot-clé « while » en Python :

Langage naturel	Python
Tant que Condition Instructions Fin Tant que	while condition: Instructions ...

Remarque — Tant que la condition est vérifiée, on recommence la boucle. La boucle s'arrête une fois que la condition n'est plus réalisée. On effectue alors les instructions qui suivent la boucle.

Exemple II.1 — Pierre dispose de 100 euros sur son compte en banque. Il compte épargner 45 euros chaque mois jusqu'à ce qu'il possède plus de 400 euros. Pour savoir dans combien de mois son objectif sera accompli, il a écrit l'algorithme suivant :

```

S ← 100
C ← 0
Tant que S < 400
| S ← S + 45
| C ← C + 1
Fin Tant que
Afficher C
  
```

1. Traduire cet algorithme en Python.

2. Exécuter cet algorithme. Pour cela, on complétera la tableau suivant en ajoutant autant de lignes que nécessaire :

S	C	S < 400 ?
...

3. Au bout de combien de mois son objectif sera-t-il rempli ?

→ À rédiger

Exemple II.2 — Écrire en langage naturel puis en Python un algorithme qui permet de trouver la plus petite puissance de 2 supérieure ou égale à un million.

→ À rédiger

Solutions

Exemple I.1

Traduction en langage Python :

```
Python
for k in range(0,11):
    a = 2*k
    print(a)
```

Cet algorithme va afficher les valeurs 0, 2, 4, 6, 8, ..., 18 et 20.

Cet algorithme sert à afficher les 11 premiers nombres pairs.

Exemple I.2

En langage naturel :

```
S ← 0
Pour K allant de 0 à 20
| S ← S + K
Fin Pour
Afficher S
```

En Python :

```
Python
s = 0
for k in range(0,21):
    s = s + k
print(s)
```

Exemple II.1

1. Traduction en Python :

```
Python
s = 100
c = 0
while s < 400:
    s = s + 45
    c = c + 1
print(c)
```

2. On a le tableau suivant :

S	C	S < 400 ?
100	0	Oui
145	1	Oui
190	2	Oui
235	3	Oui
280	4	Oui
325	5	Oui
370	6	Oui
415	7	Non

3. Son objectif sera rempli au bout de 7 mois.

Exemple II.2

En langage naturel :

```
X ← 1
Tant que X < 1 000 000
| X ← 2 * X
Fin Tant que
Afficher X
```

En langage Python :

```
Python
x = 1
while x < 1000000:
    x = 2*x
print(x)
```

Algorithmique (2ème partie)

A savoir faire à la fin du chapitre.

- Savoir ce qu'est une boucle bornée ou une boucle non bornée et comprendre les différences.
- Savoir donner le résultat d'une boucle bornée ou non bornée à l'aide d'un tableau de variables.
- Savoir programmer, dans des cas simples, une boucle bornée ou une boucle non bornée.
- Savoir lire, comprendre, modifier ou compléter un algorithme ou un programme.

Algorithmique (2ème partie)

A savoir faire à la fin du chapitre.

- Savoir ce qu'est une boucle bornée ou une boucle non bornée et comprendre les différences.
- Savoir donner le résultat d'une boucle bornée ou non bornée à l'aide d'un tableau de variables.
- Savoir programmer, dans des cas simples, une boucle bornée ou une boucle non bornée.
- Savoir lire, comprendre, modifier ou compléter un algorithme ou un programme.

Algorithmique (2ème partie)

A savoir faire à la fin du chapitre.

- Savoir ce qu'est une boucle bornée ou une boucle non bornée et comprendre les différences.
- Savoir donner le résultat d'une boucle bornée ou non bornée à l'aide d'un tableau de variables.
- Savoir programmer, dans des cas simples, une boucle bornée ou une boucle non bornée.
- Savoir lire, comprendre, modifier ou compléter un algorithme ou un programme.