Chapitre 25

Les instructions de bases de la programmation

I. Découvrir l'instruction conditionnelle

Définition 1:

La résolution de certains problèmes nécessite la mise en place d'un test pour savoir si l'on doit effectuer une instruction. Si le test est positif alors on effectue l'instruction, sinon on effectue (éventuellement) une autre instruction.

Dans un algorithme, on code la structure du « Si ... Alors ... Sinon ... » sous la forme suivante :

Si condition
  Alors Instruction 1
    Instruction 2
    ...
Sinon Instruction 1bis
    Instruction 2bis
    ...
Fin Si

Remarque :

Le « Sinon » n'est pas obligatoire. S'il n'est pas présent, aucune instruction ne sera effectuée si la condition n'est pas remplie.

Exercice 1 :

Dans l'algorithme suivant, A désigne un entier naturel :

ligne 1	Affecter à B la valeur $\sqrt{A}$
ligne 2	Affecter à C la valeur arrondie à l'unité de B
ligne 3	**Si** $B = C$
ligne 4	**Alors** Afficher " A est un carré parfait "
ligne 5	**Sinon** Afficher " A n'est pas un carré parfait "
ligne 6	**Fin Si**

1. À quelle question l'algorithme permet-il de répondre ?
2. À la fin de l'exécution de l'algorithme, quelle est la valeur de B et la valeur de C lorsque $A = 40$? Quel est le résultat affiché en sortie ?
3. Mêmes questions avec $A = 2005$.

Exercice 2 :

On considère l'algorithme suivant où A et B désignent des nombres entiers relatifs :

ligne 1	**Si** $3 \times A < B$
ligne 2	**Alors** $A \leftarrow 3 \times A$
ligne 3	**Sinon** $B \leftarrow 3 \times B$
ligne 4	**Fin Si**
ligne 5	$C \leftarrow A + B$

Faire fonctionner l'algorithme en complétant le tableau qui suit.

Valeur de A avant l'exécution de l'algorithme	6	-5	4	10	2
Valeur de B avant l'exécution de l'algorithme	-15	1	7	30	7
Test	faux				
Valeur de A après l'exécution de l'algorithme	6				
Valeur de B après l'exécution de l'algorithme	-45				
Valeur de C après l'exécution de l'algorithme	-39				

Exercice 3 :

On considère l'algorithme suivant où A, B et C désignent des nombres entiers naturels tels que $A \leqslant B < C$:

ligne 1	$X \leftarrow A^2 + B^2$
ligne 2	$Y \leftarrow C^2$
ligne 3	**Si** $X = Y$
ligne 4	**Alors** Afficher ...
ligne 5	**Sinon** Afficher ...
ligne 6	**Fin Si**

1. Recopier et compléter l'algorithme.
2. Calculer les valeurs successives de X et Y pour $A = 8$, $B = 15$ et $C = 17$.
   Quel est le résultat affiché à la sortie de l'algorithme dans ce cas ?
3. Mêmes questions avec $A = 5$, $B = 12$ et $C = 13$.
4. Donner d'autres valeurs de A, B et C qui satisfont le test de sortie de l'algorithme.

II. Programmer l'instruction conditionnelle

Syntaxe des instructions utiles dans cette fiche :

Langage naturel	Python
Quotient de la division euclidienne de A par B	`A//B`
Reste de la division euclidienne de A par B	`A%B`

Exercice 1 :

1. Expliquer le principe de l'algorithme ci-dessous. Que permet-il de faire ?

1	**Variables :**	a et c sont des entiers naturels
2		b est un nombre réel
3	**Entrée :**	Saisir a
4	**Traitement :**	Affecter à b la valeur de a/13
5		Affecter à c le quotient de la division euclidienne de a par 13
6		**Si** b=c
7		**alors** Afficher vrai
8		**Sinon**
9		Afficher faux
10		**Fin Si**

2. Ce même algorithme peut se traduire par le programme ci-dessous.

" == " permet de tester l'égalité de deux valeurs et " = " est le symbole de d'affectation.
Quelles valeurs obtient-on pour b et c lorsqu'on saisit $a = 182$ au départ ?
Qu'affiche l'algorithme en sortie dans ce cas ?

```
a=eval(input("Choisir un entier"))
b=a/13
c=a//13
if b==c:
    print("Vrai")
else:
    print("Faux")
```

3. **a.** Modifier le programme dans le but de vérifier si un nombre est divisible par 29.
   b. Les nombres 565 ; 6 785 ; 646 195 034 ; 1 970 659 794 sont-ils divisibles par 29 ?

Exercice 2 :
Écrire un programme permettant de vérifier si un nombre donné est divisible par 13 en effectuant un test sur le reste de la division de ce nombre par 13.

Exercice 3 :
Écrire et programmer un algorithme permettant de tester si entier naturel a un multiple d'un entier naturel b.

Exercice 4 :

1. Écrire un programme traduisant l'algorithme ci-dessous.

1	**Variables :**	a, b, c, m, n, x et y sont des entiers naturels
2	**Entrée :**	Saisir a
3		Saisir b
4		Saisir c
5	**Traitement :**	Affecter à m la valeur de a^2
6		Affecter à n la valeur de b^2
7		Affecter à x la valeur de $m+n$
8		Affecter à y la valeur de c^2
9		**Si** x=y
10		**alors** Afficher vrai
11		**Sinon**
12		Afficher faux
13		**Fin Si**

2. Tester ce programme pour les triplets suivants : $(3,4,5)$, $(5,12,13)$, $(6,8,10)$, $(7,24,25)$, $(8,15,17)$.

Exercice 5 :
Écrire et tester un programme qui demande en entrée à un client le montant total de ses achats.
En fonction de la somme dépensée, le programme affiche en sortie le prix à payer :

- Si la somme dépensée est strictement inférieure à 75€, il obtient 5% de remise.
- Si la somme dépensée est supérieure à 75€, il obtient 8% de remise.

III. Découvrir les boucles

1. La boucle itérative

Définition 2:

Lorsque l'on doit répéter un nombre de fois connu à l'avance la ou les mêmes tâches, on utilise une boucle itérative de la forme « Pour ... allant de ... à ... faire ... ».
Dans un algorithme, cette structure est codée de la façon suivante :

Pour variable **allant de** valeur de départ **à** valeur de fin **faire**
    Tâche 1
    Tâche 2
    ...
Fin pour

La variable utilisée dans la boucle est appelée compteur. À chaque passage dans la boucle, sa valeur est automatiquement augmentée de 1.

2. La boucle conditionnelle

Définition 3:

Lorsque l'on doit répéter un nombre de fois non déterminé à l'avance la ou les mêmes tâches, on utilise une boucle conditionnelle de la forme « Tant que ... faire ... ».
Dans un algorithme, cette structure est codée de la façon suivante :

Tant que condition **faire**
    Tâche 1
    Tâche 2
    ...
Fin Tant que

La boucle s'arrête quand la condition n'est plus remplie.

3. Quelques exercices pour comprendre

Exercice 1 : On considère les algorithmes suivants :

Algorithme 1 :

```
1  Variables :    n un réel
2  Entrée :       Saisir n
3  Traitement :   Tant que n < 50
4                     Affecter à n la valeur n + 1
5                 Fin Tant que
6  Sortie :       Afficher n
```

Algorithme 2 :

```
1  Variables :    n un réel
2  Entrée :       Saisir n
3  Traitement :   Tant que n < 50
4                     Affecter à n la valeur n + 1
5                     Afficher n
6                 Fin Tant que
```

1. Qu'affiche en sortie l'algorithme 1 pour $n = 45$? $n = 48.1$? $n = 53$?
2. Reprendre la question 1 avec l'algorithme 2.
3. Quelle valeur de n faut-il saisir pour obtenir en sortie l'affichage suivant avec l'algorithme 2 :
   44,3 45,3 46,3 47,3 48,3 49,3 50,3 ?

Exercice 2 :

```
1  Variables :    A un entier relatif
2  Entrée :       Saisir A
3  Traitement :   Pour i allant de 1 à 5
4                     Affecter à A la valeur A + 1
5                 Fin Pour
6  Sortie :       Afficher A
```

1. Qu'affiche en sortie l'algorithme ci-contre pour $A = 3$? Même question pour $A = -4$.
2. **a.** Quelle valeur de A faut-il saisir pour obtenir en sortie l'affichage : -5 ?
   b. Modifier l'algorithme pour qu'avec la valeur de A choisie dans la question précédente l'algorithme affiche en sortie : -9 -8 -7 -6 -5

IV. Programmer les boucles

Exercice 1 :
En Python, `range(3,8)` désigne la séquence des entiers n vérifiant $3 \leqslant n < 8$;
`range(5)` désigne la séquence des entiers 0, 1, …, 4.

```
for i in range(10) :
        print(i)
```

1. Tester le programme ci-contre. Qu'affiche-t-il en sortie ?
2. Écrire et tester un programme qui affiche tous les entiers inférieurs à 16.
3. Écrire et tester un programme qui affiche tous les entiers compris entre 18 et 45.

Exercice 2 :

```
n=0
while n<10:
        print(n)
        n=n+2
```

1. Tester le programme ci-contre. Qu'affiche-t-il en sortie ?
2. Écrire et tester un programme qui affiche tous les entiers pairs compris entre 18 et 45.
3. Écrire et tester un programme qui affiche tous les entiers impairs compris entre 50 et 150.

Exercice 3 :

1. On donne le programme ci-après.

```
s=0
for i in range(101):
        s=s+i
        print(s)
```

Recopier et compléter le tableau suivant par les premières valeurs prises par les variables s et i.

i	/	1	2	3										
s	0	1	3											

2. Quel problème permet de résoudre cet algorithme ?
3. **a.** En s'inspirant des programmes précédents, écrire et tester un programme permettant de calculer la somme des entiers de 34 à 145.
   b. Même question pour la somme des entiers de 67 à 456.

Exercice 4 :
On place un capital de 500 € sur un compte rémunéré à 3 % par an. Tous les ans le capital est donc multiplié par 1,03. L'algorithme ci-après, écrit en langage naturel, permet de calculer le nombre d'années au bout desquelles le capital sera doublé.

1 **Variables :** A un entier ; S est un réel
2 **Initialisation :** Affecter à S la valeur 500
3 Affecter à A la valeur 0
4 **Traitement :** **Tant que** $S < 1000$
5 Affecter à S la valeur $S \times 1{,}03$
6 Affecter à A la valeur $A + 1$
7 **Fin Tant que**
8 **Sortie :** Afficher A

1. Le programme ci-dessous traduisant l'algorithme précédent comprend une erreur. Corriger et tester le programme.

```
s=500
a=0
while s<1000:
        s=1.03*s
a=a+1
print(a)
```

2. Modifier le programme précédent de telle sorte que le capital et le taux de rémunération soient saisis en entrée. Le tester dans un nouveau contexte à décrire.