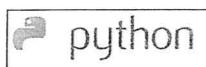


# Programmation avec le langage Python



	Il existe plusieurs logiciels permettant de programmer en langage Python. Quand on ouvre un logiciel, deux fenêtres apparaissent à l'écran : la console et l'éditeur.	
Créer un nouveau programme	On saisit le programme dans l'éditeur. On va à la ligne après chaque instruction, mais on peut taper plusieurs instructions sur la même ligne en les séparant par un point-virgule.	
Saisir A	A=input("A=") si A est une chaîne de caractères. A=float(input("A=")) si A est un flottant. A=int(input("A=")) si A est un entier. On peut utiliser la fonction <b>demande</b> avec la bibliothèque <b>lycée</b> .	
Afficher A	print(A)	
Afficher un texte, par exemple « oui »	print("oui")	
Affecter à B la valeur de A	B=A	
Écrire des commentaires	On écrit les commentaires sur une ligne, précédés de #	
Tester si $A = B$ / si $A \neq B$	A==B	A !=B
Tester si $A \geq B$ / si $A \leq B$	A>=B	A<=B
$A$ et $B$ / $A$ ou $B$	A and B	A or B
Si {condition C} Alors {instructions A} Sinon {instructions B} Fin Si	if {condition C}: {instructions A} else: {instructions B}	Il n'y a pas d'instruction de fin : c'est l'indentation (le décalage vers la droite) qui indique les instructions faisant partie de la structure conditionnelle.
Pour $i$ variant de 1 à $n$ {instructions} Fin Pour	for i in range(1,n+1): {instructions}	<ul style="list-style-type: none"> <li>L'instruction <b>for i in range(n)</b> fait parcourir à la variable <math>i</math> tous les entiers de 0 à <math>n - 1</math>.</li> <li>L'instruction <b>for i in range(<math>n_0</math>,n)</b> fait parcourir à la variable <math>i</math> tous les entiers de <math>n_0</math> à <math>n - 1</math>.</li> <li>L'instruction <b>for i in range(<math>n_0</math>,n,p)</b> fait parcourir à la variable <math>i</math> les entiers de <math>n_0</math> jusqu'à l'entier immédiatement inférieur ou égal à <math>n - 1</math> avec un pas de <math>p</math>.</li> </ul> <p>Comme pour <b>if</b>, on utilise l'indentation pour indiquer les instructions faisant partie de la boucle.</p>
Tant que {condition C} {instructions} Fin Tant que	while {condition C}: {instructions}	Comme pour <b>if</b> et <b>for</b> , on utilise l'indentation pour indiquer les instructions faisant partie de la boucle.
Définition d'une fonction $f$	<pre>def f(a,b,c,...):   y= ...   return(y)</pre>	
Racine carrée de $x$	<b>sqrt(x)</b> avec la bibliothèque <b>lycée</b> ou le module <b>math</b>	
$x$ à la puissance $n$	$x**n$	
$\pi$	<b>pi</b> avec la bibliothèque <b>lycée</b> ou le module <b>math</b>	
Reste de la division de $a$ par $b$	<b>a%b</b> ou <b>reste(a,b)</b> avec la bibliothèque <b>lycée</b>	
Nombre décimal aléatoire compris entre $a$ et $b$	<b>uniform(a,b)</b> avec la bibliothèque <b>lycée</b> ou le module <b>random</b>	
Entier aléatoire compris entre $a$ et $b$	<b>randint(a,b)</b> avec la bibliothèque <b>lycée</b> ou le module <b>random</b>	
Longueur d'une chaîne de caractères $C$	<b>len(C)</b>	
Extraire un caractère d'une chaîne de caractères $C$	$C[k]$ renvoie le $(k + 1)$ e élément de $C$	
Concaténation de deux chaînes de caractères	"abcd" + "efg" donne "abcdefg"	
Exécuter un programme	Cliquer sur ▶	

# CAHIER D'ALGORITHMIQUE

COLLECTION INDICE

NOUVELLE ÉDITION 2017

Catherine Lebert  
Michel Poncy  
Denis Vieudrin

Bordas

<b>PARTIE 1</b>	<b>Notions de base</b>
A. Algorithme – Variables – Affectation .....	3
B. Écriture d'un algorithme .....	6
C. Programmation d'un algorithme .....	10
D. Instruction conditionnelle (If) .....	14
E. Boucle bornée « POUR » (For) .....	18
F. Boucle non bornée « TANT QUE » (While) .....	22
G. Fonctions .....	26
<b>PARTIE 2</b>	<b>Fonctions numériques</b>
A. Généralités sur les fonctions – Variations .....	30
B. Fonctions de référence .....	34
<b>PARTIE 3</b>	<b>Géométrie</b>
A. Géométrie plane .....	40
B. Géométrie dans l'espace .....	44
<b>PARTIE 4</b>	<b>Statistiques et probabilités</b>
A. Statistiques .....	45
B. Échantillonnage et probabilités .....	46

La **première partie** du cahier comprend toutes les bases de l'algorithmique : variables, entrées – sorties, affectation, instruction conditionnelle, boucle bornée et boucle non bornée, fonction, ainsi que quelques éléments de programmation avec Python, Scilab et Xcas. Les exercices de cette partie ne nécessitent pas de connaissances spécifiques de la classe de Seconde.

Les **trois autres parties** proposent des exercices sur les trois grands thèmes du programme de la classe de Seconde : Fonctions numériques, Géométrie et Statistiques-Probabilités.

Chaque exercice comporte un en-tête qui décrit les notions mathématiques du programme, les compétences et les notions d'algorithmique mobilisées dans l'exercice.

Les compétences signalées sont les suivantes : **comprendre** un algorithme ; **modifier** un algorithme ; **compléter** un algorithme ; **analyser** une situation ; **écrire** un algorithme ; **programmer** un algorithme.

Ce cahier est rigoureusement conforme aux aménagements de programme de 2017 sur l'algorithmique et la programmation.

Credits iconographiques p. 24 Beboy/Fotolia ; p. 47 Renlow/Fotolia.

www. Retrouvez les corrigés des programmes demandés avec le langage Python et les logiciels Scilab et Xcas sur le site [www.bordas-indice.fr](http://www.bordas-indice.fr).

## A Algorithme – Variables – Affectation

### 1 Algorithme et notion de variable

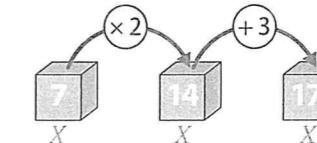
Un algorithme est une suite finie d'instructions à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes, à un certain résultat.

Pour stocker un résultat, on utilise une variable. On peut se représenter une variable comme une « boîte », un emplacement de la mémoire d'un ordinateur... Pour pouvoir accéder à son contenu, on lui donne un nom.

#### Exemple 1 : Un programme de calcul

Choisir le nombre 7  
Multiplier ce nombre par 2  
Ajouter 3 au résultat

On peut ici utiliser une variable, que l'on nomme *X*, qui contient les résultats obtenus à la fin de chaque étape du programme de calcul.



Une variable est désignée par un nom.

Elle contient une valeur. On utilise cette année trois types de valeurs :

- entier (nombre entier relatif) ;
- flottant (nombre à virgule) ;
- chaîne de caractères : suite ordonnée de caractères, un caractère étant un chiffre, une lettre, un symbole... .

#### Remarques :

1) « Année » est une chaîne de cinq caractères. On dit que c'est une chaîne de longueur 5.

Le premier caractère de cette chaîne est *A*, son deuxième caractère est *n*, son troisième est *n*...

2) Les chaînes de caractères peuvent « s'ajouter », c'est-à-dire se mettre bout-à-bout (on parle aussi de concaténation de chaînes de caractères). Par exemple, « Année » + « 2017 » = « Année2017 ».

### 2 L'affectation

Lorsque l'on donne une valeur à une variable *X*, on écrit l'une des instructions ci-dessous :

Affecter à *X* la valeur ..... ou *X* prend la valeur .....

La nouvelle valeur remplace la valeur précédente.

Suite d'instructions correspondant au programme de calcul de l'exemple 1 :

La variable *X* contient d'abord 7. → *X* prend la valeur 7  
La valeur de *X* est ensuite multipliée par 2 : elle devient donc  $2X$ . → *X* prend la valeur  $2X$   
Elle est enfin augmentée de 3 : elle devient donc  $X + 3$ . → *X* prend la valeur  $X + 3$

#### Exemple 2 : une suite d'instructions

Suite d'instructions :

	<i>A</i>	<i>B</i>	Valeur de la variable <i>A</i> et valeur de la variable <i>B</i> après l'exécution de chaque instruction :
A prend la valeur 3	3		← La valeur de <i>A</i> est 3 et <i>B</i> n'a pas encore de valeur.
B prend la valeur <i>A</i> + 1	3	4	← Comme la valeur de <i>A</i> est 3, la valeur de <i>B</i> est : $3 + 1$ soit 4. La valeur de <i>A</i> ne change pas : elle reste égale à 3.
A prend la valeur <i>A</i> + <i>B</i>	7	4	← Comme la valeur de <i>A</i> est 3 et que celle de <i>B</i> est 4, la nouvelle valeur de <i>A</i> est : $3 + 4$ , c'est-à-dire 7. La valeur de <i>B</i> ne change pas : elle reste égale à 4.

**EXERCICE 1**

On donne deux suites d'instructions  $S_1$  et  $S_2$ , où la variable  $A$  est une chaîne de caractères.

 $S_1$ 

A prend la valeur « bonjour »  
 A prend la valeur « bonsoir »

 $S_2$ 

A prend la valeur « bonsoir »  
 A prend la valeur « bonjour »

1. a. Après l'exécution des instructions de  $S_1$ , la valeur de la variable  $A$  est : ....bonsoir.....
  - b. Après l'exécution des instructions de  $S_2$ , la valeur de la variable  $A$  est : ....bonjour.....
  2. L'ordre dans lequel on écrit des instructions a-t-il de l'importance ?
- Oui, l'ordre dans lequel on écrit des instructions a de l'importance.....

**EXERCICE 2**

Soit deux variables  $A$  et  $B$ .

Justifier qu'à l'issue des instructions ci-dessous, la valeur de la variable  $B$  est égale à 12.

A prend la valeur 1  
 B prend la valeur 3  
 A prend la valeur  $A + B$   
 B prend la valeur  $A \times B$

À l'issue de la 3<sup>e</sup> instruction, la valeur de  $A$  est  $1 + 3$ , soit 4, et la valeur de  $B$  est 3.....  
À l'issue de la dernière instruction, la valeur de  $B$  est  $4 \times 3$ , soit 12.....

**EXERCICE 3**

1. a. Joe a écrit la valeur de la variable  $X$  après l'exécution de chaque instruction ci-dessous.

Deux valeurs sont fausses. Corriger son travail.

Affecter à  $X$  la valeur 4  
 Affecter à  $X$  la valeur  $2X$   
 Affecter à  $X$  la valeur  $X + 3$   
 Affecter à  $X$  la valeur  $X \times X$

$X$	Correction
4	4
8	8
$7 \leftarrow$ faux	$\dots 8 + 3 = 11 \dots$
$16 \leftarrow$ faux	$\dots 11 \times 11 = 121 \dots$

- b. On remplace la première instruction par « Affecter à  $X$  la valeur  $a$  »,  $a$  étant un réel donné.

Quelle est, parmi les valeurs ci-dessous, celle de la variable  $X$  après l'exécution de ces instructions ?

$2a + 3^2$         $(2a + 3)^2$         $2(a + 3)^2$

2. a. Écrire la valeur de la variable  $X$  après l'exécution de chaque instruction ci-dessous.

$X$
4
$\dots 4 + 3 = 7 \dots$
$\dots 7 \times 7 = 49 \dots$
$\dots 2 \times 49 = 98 \dots$

- b. On remplace la première instruction par « Affecter à  $X$  la valeur  $a$  »,  $a$  étant un réel donné.

Quelle est la valeur de la variable  $X$ , après l'exécution de ces instructions ?

La valeur de  $X$  est  $2(a + 3)^2$ .....

**EXERCICE 4**

On donne ci-contre une suite d'instructions.

1. Quelles sont les variables utilisées ?

Les variables sont  $A$ ,  $B$  et  $P$ .....

2. a. Déterminer la valeur de  $P$  après l'exécution de la dernière instruction.

Après la troisième instruction, la valeur de  $P$  est  $10 + 15$ , soit 25.....

Après la dernière instruction, la valeur de  $P$  est  $2 \times 25$ , soit 50.....

- b. On considère un rectangle de largeur 10 cm et de longueur 15 cm.  
À quoi correspond cette valeur de  $P$  ?

La valeur de  $P$  correspond au périmètre du rectangle.....

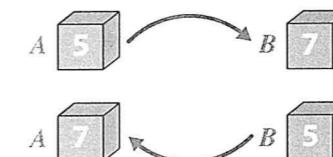
3. Modifier la dernière instruction afin que la valeur de  $P$ , après l'exécution de cette instruction, soit égale à la moyenne des nombres 10 et 15.

On remplace la dernière instruction par « Affecter à  $P$  la valeur  $\frac{P}{2}$  ».....

Affecter à  $A$  la valeur 10  
Affecter à  $B$  la valeur 15  
Affecter à  $P$  la valeur  $A + B$   
Affecter à  $P$  la valeur  $2P$

**EXERCICE 5**

On considère deux variables  $A$  et  $B$  qui ont respectivement pour valeurs 5 et 7. On souhaite échanger les valeurs de  $A$  et de  $B$ .



1. a. Dans chacun des cas suivants, écrire la valeur de la variable  $A$  et celle de la variable  $B$  après l'exécution de chaque instruction.

$A$	$B$
5	7
7	7

$A$	$B$
5	5
5	5

- b. Ces deux suites d'instructions permettent-elles d'échanger les valeurs de  $A$  et de  $B$  ?

Non, ces deux suites d'instructions ne le permettent pas.....

2. On crée une nouvelle variable :  $C$ .

Compléter la suite d'instructions ainsi que le tableau ci-dessous, afin qu'après l'exécution de la dernière instruction, la valeur de  $A$  soit égale à 7 et que celle de  $B$  soit égale à 5.

$A$	$B$	$C$
5	7	5
7	7	5
7	5	5

## B Écriture d'un algorithme

### 1 Structure d'un algorithme

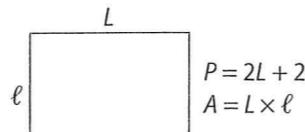
Avant de commencer à écrire un algorithme, on analyse le problème et on crée les variables que l'on va utiliser en leur donnant un nom et en précisant leur type (entier, flottant, chaîne de caractères) : c'est ce que l'on appelle la **déclaration des variables**.

Un algorithme comprend ensuite trois phases :

- une phase d'**initialisation** qui permet de donner une valeur initiale aux variables ;
- une phase de **traitement** du problème ;
- une phase de **sortie** des résultats.

#### Exemple : périmètre et aire d'un rectangle

On souhaite écrire un algorithme qui calcule et affiche le périmètre (en cm) et l'aire (en  $\text{cm}^2$ ) d'un rectangle de longueur 4 cm et de largeur 3 cm.



On crée quatre variables  $L$ ,  $\ell$ ,  $P$  et  $A$  qui contiendront respectivement la longueur, la largeur, le périmètre et l'aire du rectangle.

La longueur est 4 cm : on donne à  $L$  la valeur 4.      } → **Initialisation**  
La largeur est 3 cm : on donne à  $\ell$  la valeur 3.

On calcule le périmètre : on donne à  $P$  la valeur  $2L + 2\ell$ .      } → **Traitement**  
On calcule l'aire : on donne à  $A$  la valeur  $L \times \ell$ .

On affiche le périmètre et l'aire du rectangle.      } → **Sortie**

On pourrait aussi calculer et afficher le périmètre et l'aire d'un rectangle pour n'importe quelles longueur et largeur en demandant à l'utilisateur d'entrer les valeurs de  $L$  et de  $\ell$ .

### 2 Instructions d'Entrée-Sortie

#### Instructions d'Entrée

Pour entrer une donnée, on écrit l'une des instructions ci-dessous :

Saisir  $X$  ou Demander la valeur de  $X$   
ou Lire  $X$

Ces instructions affectent à la variable  $X$  la valeur saisie par l'utilisateur.

Voici l'algorithme de l'exemple précédent :

Variables	$L$ , $\ell$ , $P$ et $A$ sont des flottants
Initialisation	$L$ prend la valeur 4 $\ell$ prend la valeur 3
Traitement	$P$ prend la valeur $2L + 2\ell$ $A$ prend la valeur $L \times \ell$
Sortie	Afficher $P$ et $A$

Si on demande à l'utilisateur d'entrer les valeurs de  $L$  et de  $\ell$ , on remplace le terme « **Initialisation** » par « **Entrée** », comme ci-dessous.

Variables	$L$ , $\ell$ , $P$ et $A$ sont des flottants
Entrée	Saisir $L$ et $\ell$
Traitement	$P$ prend la valeur $2L + 2\ell$ $A$ prend la valeur $L \times \ell$
Sortie	Afficher $P$ et $A$

#### Instruction de Sortie

Pour afficher un résultat, on écrit l'instruction :

Afficher ...

L'instruction Afficher  $X$  affiche la valeur contenue dans la variable  $X$ .

L'instruction Afficher « ..... » affiche le texte écrit entre les guillemets.

Cet algorithme affiche en sortie :

$P = 14$  et  $A = 12$ .

## EXERCICE 6 Moyenne

### Compléter un algorithme

Pour ce trimestre, Suzanne a obtenu trois notes en mathématiques, désignées par les variables  $A$ ,  $B$  et  $C$ .

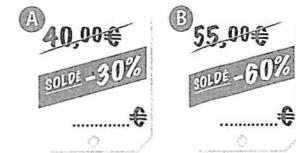
Compléter l'algorithme ci-dessous afin qu'il permette à Suzanne de connaître sa moyenne du trimestre.

Variables	$A$ , $B$ , $C$ et $M$ sont des flottants
Entrée	Saisir .... $A$ , $B$ et $C$
Traitement	$M$ prend la valeur $\frac{A + B + C}{3}$
Sortie	Afficher .... $M$

## EXERCICE 7 Pourcentages

### Comprendre un algorithme

En prévision des soldes, un commerçant s'apprête à modifier ses étiquettes.



1. Calculer le prix que doit inscrire le commerçant sur l'étiquette A.

$$40 - 40 \times \frac{30}{100} = 40 - 12 = 28$$

Le commerçant doit inscrire 28 euros.

2. Voici un algorithme.

Variables	$P$ et $T$ sont des flottants
Entrée	Demander les valeurs de $P$ et $T$
Traitement	$P$ prend la valeur $P - P \times \frac{T}{100}$
Sortie	Afficher $P$

a. Quelle est la valeur affichée en sortie lorsque l'on entre  $P = 40$  et  $T = 30$  ?

$$40 - 40 \times \frac{30}{100} = 28. La valeur affichée est 28.$$

b. Quel est le rôle de cet algorithme ?

Cet algorithme affiche le prix d'un article soldé, lorsque l'on entre l'ancien prix et la remise effectuée.

c. Que doit entrer le commerçant afin de compléter l'étiquette B ?

Le commerçant doit entrer  $P = 55$  et  $T = 60$ .

## EXERCICE 8 Chaînes de caractères

### Comprendre un algorithme

On donne ci-dessous un algorithme.

Variables	$C$ , $C1$ , $C2$ , $C3$ et $S$ sont des chaînes de caractères
Entrée	Saisir $C$
Traitement	Affecter à $C1$ la valeur du 1 <sup>er</sup> caractère de $C$ Affecter à $C2$ la valeur du 2 <sup>e</sup> caractère de $C$ Affecter à $C3$ la valeur du 3 <sup>e</sup> caractère de $C$ Affecter à $S$ la valeur $C3 + C2 + C1$
Sortie	Afficher $S$

Qu'affiche cet algorithme lorsque l'on saisit MOT ?

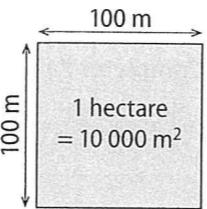
$C1$ .prend la valeur M,  $C2$ .prend la valeur O et  $C3$ .prend la valeur T.

Donc  $S$ .prend la valeur TOM. La valeur affichée est celle de S, c'est-à-dire TOM.

**EXERCICE****Aire****Compléter un algorithme**

Compléter chacun des algorithmes ci-dessous afin qu'ils affichent en  $m^2$ , puis en hectares, la superficie d'un terrain rectangulaire de longueur  $L$  mètres et de largeur  $\ell$  mètres, saisies en entrée.

<b>Variables</b>	$M, H, \ell$ et $L$ sont des flottants
<b>Entrée</b>	Saisir $\ell$ et $L$
<b>Traitement</b>	$M$ prend la valeur $L \times \ell$ $H$ prend la valeur $\frac{M}{10\,000}$
<b>Sortie</b>	Afficher $M$ et $H$



<b>Variables</b>	$M, \ell$ et $L$ sont des flottants
<b>Entrée</b>	Saisir $\ell$ et $L$
<b>Traitement et sortie</b>	$M$ prend la valeur $L \times \ell$ Afficher $M$ $M$ prend la valeur $\frac{M}{10\,000}$ Afficher $M$

**EXERCICE****Proportionnalité****Comprendre un algorithme • Écrire un algorithme**

Oleg s'apprête à préparer un gratin de pommes de terre pour 6 personnes à l'aide de la recette ci-contre.

1. Calculer la quantité nécessaire de pommes de terre pour 6 personnes.

$6 \times 1/4 = 1,5$  : il faut 1,5 kg de pommes de terre.

2. On donne l'algorithme ci-dessous.

<b>Variable</b>	$Q$ est un flottant
<b>Entrée</b>	Demander la valeur de $Q$
<b>Traitement</b>	$Q$ prend la valeur $6 \times \frac{Q}{4}$
<b>Sortie</b>	Afficher $Q$

- Ingrédients (pour 4 personnes)
- 1 kg de pommes de terre
  - 60 g de fromage râpé
  - 25 cl de crème fraîche



a. Quel est le rôle de cet algorithme ?

Cet algorithme affiche, lorsque l'on entre une quantité pour 4 personnes, la quantité nécessaire pour 6 personnes.

b. Que doit saisir Oleg pour connaître la quantité de crème fraîche (en cl) nécessaire pour 6 personnes ?

Oleg doit saisir 25.

3. Écrire un algorithme qui affiche, lorsque l'on entre la quantité  $Q$  nécessaire pour  $N$  personnes, la quantité qu'il faudra pour  $M$  personnes.

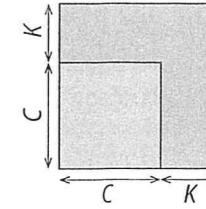
<b>Variables</b>	$Q, N$ et $M$ sont des flottants
<b>Entrée</b>	Demandez les valeurs de $Q, N$ et $M$
<b>Traitement</b>	$Q$ prend la valeur $M \times \frac{Q}{N}$
<b>Sortie</b>	Afficher $Q$

**EXERCICE****Périmètre et aire****Comprendre un algorithme • Modifier un algorithme**

On considère un carré de côté  $C$  (en centimètres) que l'on agrandit en ajoutant  $K$  centimètres à chacun de ses côtés.

On donne l'algorithme ci-dessous :

<b>Variables</b>	$A, B, C, D$ et $K$ sont des flottants
<b>Initialisation</b>	$C$ prend la valeur 10
<b>Entrée</b>	Saisir $K$
<b>Traitement</b>	$A$ prend la valeur $4 \times C$ $B$ prend la valeur $C + K$ $D$ prend la valeur $4 \times B$
<b>Sortie</b>	Afficher $A$ et $D$



1. a. Quelle est, en centimètres, la mesure du côté du carré initial ?

Le côté du carré initial mesure 10 cm.

b. Quelles sont les valeurs affichées en sortie lorsque l'on entre  $K = 2$  ?

L'algorithme affiche 40 et 48.

c. Quel est le rôle de cet algorithme ?

L'algorithme affiche, lorsque l'on entre la valeur de  $K$ , le périmètre du carré initial et celui du carré agrandi.

2. Quelles lignes faut-il modifier afin que soient affichées l'aire du carré initial ainsi que l'aire du carré agrandi ?

On remplace «  $A$  prend la valeur  $4 \times C$  » par «  $A$  prend la valeur  $C \times C$  » et «  $D$  prend la valeur  $4 \times B$  » par «  $D$  prend la valeur  $B \times B$  ».

**EXERCICE****Vitesse moyenne****Compléter un algorithme**

On rappelle la formule :  $D = V \times T$  où  $V$  est la vitesse moyenne (en km/h),  $D$  la distance parcourue (en km) et  $T$  la durée du parcours (en h).

1. Aïda doit parcourir 572 km.

Calculer la durée (en heures) de son parcours si sa vitesse moyenne est de :

- a. 110 km/h ; b. 130 km/h.

a.  $T = 572 / 110 = 5,2$ . La durée est de 5 heures et 12 minutes, car  $0,2 \text{ h} = 0,2 \times 60 \text{ min} = 12 \text{ min}$ .

b.  $T = 572 / 130 = 4,4$ . La durée est de 4 heures et 24 minutes, car  $0,4 \text{ h} = 0,4 \times 60 \text{ min} = 24 \text{ min}$ .

2. Compléter l'algorithme ci-dessous afin que soient affichées (en heures) pour un trajet de  $D$  kilomètres les durées  $T1$  et  $T2$  du parcours lorsqu'on roule respectivement à 110 km/h et à 130 km/h, ainsi que la différence entre ces deux durées.

<b>Variables</b>	$D, T1, T2$ et $T$ sont des flottants
<b>Entrée</b>	Saisir $D$
<b>Traitement</b>	$T1$ prend la valeur $\frac{D}{110}$ $T2$ prend la valeur $\frac{D}{130}$ $T$ prend la valeur $T1 - T2$
<b>Sortie</b>	Afficher $T1, T2$ et $T$

# C Programmation d'un algorithme

## 1 La programmation

Les algorithmes étudiés dans les pages précédentes peuvent être programmés sur un ordinateur en utilisant un langage de programmation adapté.

**Programmer un algorithme**, c'est le traduire dans un langage compréhensible par un logiciel donné.

Nous abordons dans cette page le langage de programmation Python et ceux des logiciels Scilab et Xcas.

## 2 Les commandes pour démarrer

Les commandes indispensables pour élaborer un programme sont les suivantes :

- les commandes permettant de créer un programme à partir de l'algorithme déjà fait ;
- les instructions d'entrée et de sortie des données ;
- l'instruction d'affectation ;
- les commandes permettant d'exécuter le programme réalisé.

	Python	Scilab	Xcas
Créer un nouveau programme	Cliquer sur <u>Nouveau Fichier</u> , ou, dans le menu <u>Fichier</u> , choisir <u>Nouveau</u> puis <u>Nouveau Fichier</u>	Dans le menu <u>Applications</u> , choisir <u>SciNotes</u> , puis ouvrir l'éditeur	Dans le menu <u>Prg</u> , choisir <u>Nouveau programme</u>
Saisir A	A= <b>input</b> ("A=") si A est une chaîne de caractères A= <b>int(input("A="))</b> si A est un entier A= <b>float(input("A="))</b> si A est un nombre (flottant)	A= <b>input</b> ("A=")	<b>input(A)</b> ou <b>saisir(A)</b>
Afficher A	<b>print(A)</b>	<b>disp(A)</b> ou avec le module « lycée » : <b>afficher(A)</b>	<b>print(A)</b> ou <b>afficher(A)</b>
Afficher un texte, par exemple « oui »	<b>print("oui")</b>	<b>disp("oui")</b>	<b>print("oui")</b> ou <b>afficher("oui")</b>
Affecter à B la valeur de A	B=A	B=A	B:=A
Obtenir la longueur d'une chaîne de caractères C	<b>len(C)</b>	<b>length(C)</b>	<b>dim(C)</b>
Extraire un caractère d'une chaîne de caractères C	<b>C[k]</b> renvoie le $(k+1)^{\text{e}}$ élément de C	<b>part(C, k)</b> renvoie le $k^{\text{e}}$ élément de C	<b>C[k]</b> renvoie le $(k+1)^{\text{e}}$ élément de C
Concaténation de deux chaînes de caractères	"abcd" + "efg" donne "abcdefg"	"abcd" + "efg" donne "abcdefg"	"abcd" + "efg" donne "abcdefg"
Exécuter un programme	Cliquer sur	Cliquer sur  , ou sur  pour enregistrer puis exécuter	Cliquer sur

### Remarque :

En langage Python, lorsque l'on utilise l'instruction `A=input("A=")`, la valeur de la variable A est une chaîne de caractères (type `str`). Si la valeur attendue de A est un entier (type `int`), l'instruction précédente doit être suivie de l'instruction `A=int(A)` ; si la valeur attendue de A est un réel (type `float`), l'instruction précédente doit être suivie de l'instruction `A=float(A)`. Avec la fonction `demande` de la bibliothèque « lycée », le type de la variable A est reconnu automatiquement.

```
N=input("N=")
N=int(N)
G=input("G=")
G=float(G)
M=G/N
print("Gain par personne")
print(M)
```

## EXERCICE 13

Comprendre un algorithme • Programmer un algorithme

1. Qu'affiche en sortie l'algorithme ci-contre pour les valeurs suivantes de la variable A saisies en entrée :

- a.  $A = 5$  ?  
b.  $A = -9$  ?  
a. L'affichage est 30.....  
b. L'affichage est 86.....

Variables	A et B sont des entiers
Entrée	Demander la valeur de A
Traitement	B prend la valeur $A^2 + 5$
Sortie	Afficher B

2. Programmer cet algorithme. [www.](http://www.)

Python	Scilab	Xcas
A=int(input("A="))	A=input("A=")	saisir(A);
B=A**2+5	B=A^2+5	B:=A^2+5;
print(B)	disp(B)	afficher(B);

## EXERCICE 14

Comprendre un programme

On considère le programme ci-dessous écrit à l'aide de trois langages de programmation.

Python	Scilab	Xcas
X=float(input("X=")) Y=float(input("Y=")) Z=X X=Y Y=Z print(X) print(Y)	X=input("X=") Y=input("Y=") Z=X X=Y Y=Z disp(X) disp(Y)	saisir(X); saisir(Y); Z:=X; X:=Y; Y:=Z; afficher(X); afficher(Y);

1. Qu'affiche ce programme en sortie pour les valeurs suivantes des variables X et Y saisies en entrée :

- a.  $X = 5$  et  $Y = 9$  ?  
b.  $X = 0,7$  et  $Y = 1,8$  ?  
a. Il affiche  $X = 9$  et  $Y = 5$ .....  
b. Il affiche  $X = 1,8$  et  $Y = 0,7$ .....

2. Que fait ce programme ?

Il échange les valeurs des variables X et Y.....

## EXERCICE 15

Écrire un algorithme

On a réalisé un programme à l'aide de trois langages de programmation.

Python	Scilab	Xcas
A=float(input("A=")) B=float(input("B=")) C=A-B D=A+B E=C*D+B**2 print(E)	A=input("A=") B=input("B=") C=A-B D=A+B E=C*D+B^2 disp(E)	saisir(A); saisir(B); C:=-A-B; D:=A+B; E:=C*D+B^2; afficher(E)

Écrire ci-dessous l'algorithme correspondant en mettant en évidence les différentes étapes de l'algorithme.

Variables	A, B, C, D, E sont des flottants.....
Entrée	Saisir A, B.....
Traitement	C prend la valeur A - B..... D prend la valeur A + B..... E prend la valeur C x D + B <sup>2</sup> .....
Sortie	Afficher E.....

**EXERCICE 16**

Aire et périmètre

Compléter et modifier un programme

On donne ci-dessous, le programme incomplet écrit dans trois langages de programmation.

Python	Scilab	Xcas
<code>C=float(input("C="))</code>	<code>C=input("C=")</code>	<code>saisir(C);</code>
<code>D=float(input("D="))</code>	<code>D=input("D=")</code>	<code>saisir(D);</code>
<code>A=C*D/2.....</code>	<code>A=C*D/2.....</code>	<code>A:=C*D/2.....;</code>
<code>print(A)</code>	<code>disp(A)</code>	<code>afficher(A)</code>

1. Compléter ce programme pour qu'il affiche l'aire  $A$  d'un triangle rectangle connaissant les deux côtés de l'angle droit.

2. Modifier ce programme pour qu'il affiche également le périmètre de ce triangle.

Python	Scilab	Xcas
<code>C=float(input("C="))</code>	<code>C=input("C=")</code>	<code>saisir(C);</code>
<code>D=float(input("D="))</code>	<code>D=input("D=")</code>	<code>saisir(D);</code>
<code>A=C*D/2.....</code>	<code>A=C*D/2.....</code>	<code>A:=C*D/2.....;</code>
<code>P=C+D+sqrt(C**2+D**2)</code>	<code>P=C+D+sqrt(C.^2+D.^2)</code>	<code>P:=C+D+sqrt(C.^2+D.^2);</code>
<code>print(A)</code>	<code>disp(A)</code>	<code>afficher(A);</code>
<code>print(P)</code>	<code>disp(P)</code>	<code>afficher(P);</code>

**EXERCICE 17**

Compléter un algorithme • Programmer un algorithme

Le droit d'entrée journalier dans un parc aquatique est 37 € pour un adulte et 28 € pour un enfant.

L'algorithme ci-dessous permet de calculer le prix payé par un groupe comprenant des adultes et des enfants.

Les variables  $A$  et  $E$  représentent le nombre d'adultes et le nombre d'enfants du groupe et la variable  $P$  représente le prix payé par le groupe.

Variables	$A$ , $E$ et $P$ sont des entiers
Entrée	Demander les valeurs de $A$ et $E$
Traitement	$P$ prend la valeur $37 \times A + 28 \times E$
Sortie	Afficher « Le prix est : » Afficher $P$

1. Compléter cet algorithme.

2. Programmer cet algorithme. [www.](#)

Python	Scilab	Xcas
<code>A=int(input("A="))</code>	<code>A=input("A=")</code>	<code>saisir(A);</code>
<code>E=int(input("E="))</code>	<code>E=input("E=")</code>	<code>saisir(E);</code>
<code>P=37*A+28*E</code>	<code>P=37*A+28*E</code>	<code>P:=37*A+28*E;</code>
<code>print("Le prix est")</code>	<code>disp("Le prix est")</code>	<code>afficher("Le prix est");</code>
<code>print(P)</code>	<code>disp(P)</code>	<code>afficher(P);</code>

**EXERCICE 18**

Calcul algébrique

Programmer un algorithme • Analyser une situation

On considère l'algorithme ci-dessous.

<b>Variables</b>	$N$ est un entier non nul
<b>Entrée</b>	$A$ et $B$ sont des flottants
<b>Traitement</b>	Demander la valeur de $N$
	$A$ prend la valeur $\frac{2}{N}$
	$B$ prend la valeur $\frac{1}{N} + \frac{1}{2N} + \frac{1}{3N} + \frac{1}{6N}$
<b>Sortie</b>	Afficher $A$ , $B$

1. Programmer cet algorithme. [www.](#)

Python	Scilab	Xcas
<code>N=int(input("N="))</code>	<code>N=input("N=")</code>	<code>saisir(N);</code>
<code>A=2/N</code>	<code>A=2/N</code>	<code>A:=2/N;</code>
<code>B=1/N+1/(2*N)+1/(3*N)+1/(6*N)</code>	<code>B=1/N+1/(2*N)+1/(3*N)+1/(6*N)</code>	<code>B:=1/N+1/(2*N)+1/(3*N)+1/(6*N);</code>
<code>print(A)</code>	<code>disp(A)</code>	<code>afficher(A);</code>
<code>print(B)</code>	<code>disp(B)</code>	<code>afficher(B);</code>

2. Faire fonctionner ce programme pour diverses valeurs de  $N$ . Que constate-t-on ? Le démontrer.

On constate que  $A$  et  $B$  semblent égaux.Pour démontrer cette conjecture, on peut simplifier  $B$  :

$$B = \frac{1}{N} + \frac{1}{2N} + \frac{1}{3N} + \frac{1}{6N} = \frac{6}{6N} + \frac{3}{6N} + \frac{2}{6N} + \frac{1}{6N} = \frac{12}{6N} = \frac{2}{N} = A.$$

**EXERCICE 19**

Calcul algébrique

Comprendre un programme • Analyser une situation

On considère le programme suivant écrit dans trois langages.

Python	Scilab	Xcas
<code>N=int(input("N="))</code>	<code>N=input("N=")</code>	<code>saisir(N);</code>
<code>P=N+1</code>	<code>P=N+1</code>	<code>P:=N+1;</code>
<code>Q=P+1</code>	<code>Q=P+1</code>	<code>Q:=P+1;</code>
<code>R=Q+1</code>	<code>R=Q+1</code>	<code>R:=Q+1;</code>
<code>S=N**2-P**2-Q**2+R**2</code>	<code>S=N**2-P^2-Q^2+R^2</code>	<code>S:=-N^2-P^2-Q^2+R^2;</code>
<code>print(S)</code>	<code>disp(S)</code>	<code>afficher(S)</code>

1. Faire fonctionner ce programme pour diverses valeurs de  $N$ . Quelle conjecture peut-on faire sur  $S$  ?

On conjecture que  $S$  vaut 4 quelle que soit la valeur de  $N$  saisie.

2. Démontrer la conjecture faite sur  $S$ .

On a  $P = N + 1$  donc  $Q = P + 1 = N + 2$  et  $R = Q + 1 = N + 3$ .D'où  $S = N^2 - (N+1)^2 - (N+2)^2 + (N+3)^2 = N^2 - (N^2 + 2N + 1) - (N^2 + 4N + 4) + (N^2 + 6N + 9)$ ,Soit  $S = N^2 - N^2 - 2N - 1 - N^2 - 4N - 4 + N^2 + 6N + 9 = 4$ .

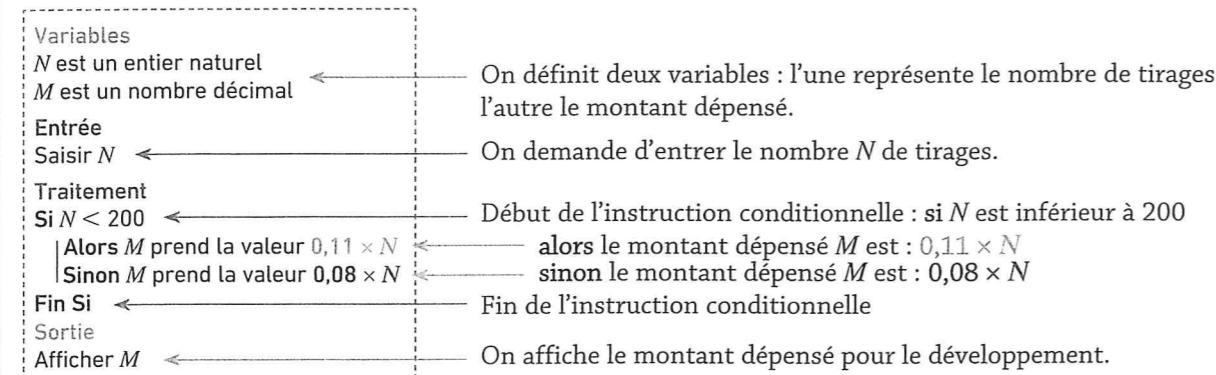
# D Instruction conditionnelle (If)

## 1 Structure d'une instruction conditionnelle

### Exemple : développement de photos

Un site internet de développement de photos propose le tirage sur papier des photos au tarif de 0,11 € l'unité ; le tarif passe à 0,08 € l'unité pour une commande d'au moins 200 photos.  
On veut créer un algorithme donnant le montant dépensé pour un nombre  $N$  de tirages.  
Pour cela, on doit introduire une instruction conditionnelle dans l'algorithme :

- si le nombre de photos  $N$  est strictement inférieur à 200, le montant dépensé est  $N \times 0,11$ , puisque le prix d'une photo est alors 0,11 € ;
- si le nombre de photos  $N$  est supérieur ou égal à 200, le montant dépensé est  $N \times 0,08$ , puisque le prix d'une photo est alors 0,08 €.



C'est la structure alternative « Si... Alors... Sinon... » qui permet d'écrire l'**instruction conditionnelle** dans l'algorithme : un test est effectué sur une condition  $C$ , et le résultat du test décide de l'exécution de la phase de traitement.

Si {condition C}	Si la condition $C$ est vérifiée, seules les instructions $A$ sont exécutées.
Alors {instructions A}	Si la condition $C$ n'est pas vérifiée, seules les instructions $B$ sont exécutées.
Sinon {instructions B}	
Fin Si	

**Remarque :** on peut aussi utiliser la structure incomplète : « Si... Alors... » ; dans ce cas, si la condition  $C$  n'est pas vérifiée, l'exécution de l'algorithme continue après le **Fin Si**.

## 2 Programmation

### Programmation d'une instruction conditionnelle :

	Python	Scilab	Xcas
Si {condition C} Alors {instructions A} Sinon {instructions B} Fin Si	if {condition C} : {instructions A} else : {instructions B}	if {condition C} then {instructions A} else {instructions B} end	si {condition C} alors {instructions A} sinon {instructions B} fsi ;

### Programmes relatifs à l'algorithme vu en exemple :

Python	Scilab	Xcas
<pre> N=int(input("entrer N:")) if N&lt;200:     M=0.11*N else:     M=0.08*N print(M) </pre>	<pre> N=input("entrer N:") si N&lt;200     then M=0.11*N     else M=0.08*N end disp(M) </pre>	<pre> saisir("entrer N:",N) si N&lt;200     alors M:=0.11*N     sinon M:=0.08*N     fsi; afficher(M) </pre>

## EXERCICE 20

### Comprendre un algorithme

On considère l'algorithme ci-contre.

Qu'affiche cet algorithme en sortie pour les valeurs suivantes de la variable  $X$  saisies en entrée ?

- |             |               |
|-------------|---------------|
| a. $X = 17$ | b. $X = 108$  |
| c. $X = 87$ | d. $X = 2786$ |
- a. Si  $X = 17$ , alors  $Y = 14$ .  
b. Si  $X = 108$ , comme 108 est multiple de 3, alors  $Y = 36$ .  
c. Si  $X = 87$ , comme 87 est multiple de 3, alors  $Y = 29$ .  
d. Pour  $X = 2786$ , alors  $Y = 2783$ , car 2786 n'est pas multiple de 3.

Variables	$X$ et $Y$ sont des entiers
Entrée	Saisir $X$
Traitement	Si $X$ est un multiple de 3 Alors $Y$ prend la valeur $\frac{X}{3}$ Sinon $Y$ prend la valeur $X - 3$
Sortie	Fin Si Afficher $Y$

## EXERCICE 21

### Triangle rectangle

### Compléter un algorithme

On souhaite que le programme incomplet donné ci-dessous en langage Python affiche, après saisie des longueurs des trois côtés d'un triangle dont on connaît le plus grand côté, si ce triangle est ou non rectangle.

Compléter ce programme.

```

a=float(input("Plus grand côté?"))
b=float(input("Second côté?"))
c=float(input("Troisième côté?"))
if a**2==b**2+c**2:
    print("Le triangle est rectangle")
else:
    print("Le triangle n'est pas rectangle")
  
```

## EXERCICE 22

### Compléter un algorithme • Modifier un algorithme

Un groupe de personnes souhaite réserver un chalet pour les sports d'hiver. Le prix de la location à la semaine est 800 €.

Le forfait pour skier toute la semaine est de 220 € par personne, mais il existe un tarif « groupe » à 180 € par personne à partir de 5 personnes d'un même groupe.

Variables	$N$ et $P$ sont des entiers
Entrée	Lire $N$
Traitement	Si $N \geq 5$ Alors $P$ prend la valeur $800 + 180 \times N$ Sinon $P$ prend la valeur $800 + 220 \times N$
Sortie	Fin Si Afficher $P$

1. a. Quel est le prix payé par un groupe de 4 personnes ?

$800 + 220 \times 4$ , soit 1 680 €.

- b. Quel est le prix payé par un groupe de 6 personnes ?

$800 + 180 \times 6$ , soit 1 880 €.

2. Compléter l'algorithme ci-dessus afin qu'il affiche en sortie le prix payé par le groupe pour la semaine, selon le nombre de personnes du groupe saisi en entrée.

3. Comment doit-on modifier cet algorithme afin qu'il affiche en sortie le prix que doit payer chaque membre du groupe ?

On définit une nouvelle variable réelle  $Q$ .

Après **Fin Si**, on ajoute l'instruction : «  $Q$  prend la valeur  $\frac{P}{N}$  ».

À la place de « **Afficher  $P$**  », on écrit : « **Afficher  $Q$**  ».

**EXERCICE 23**

Compléter un algorithme • Programmer un algorithme

Lorsque c'est l'été en France, il y a 5 heures de décalage avec le Brésil, c'est-à-dire qu'il est 5 h à Rio de Janeiro quand il est 10 h à Paris.

1. Quelle heure est-il à Rio :

- a. quand il est 19 h à Paris ?  
b. quand il est 2 h à Paris ?  
Il est 14 h à Rio.  
Il est 21 h à Rio.

2. Compléter l'algorithme ci-dessous afin qu'il affiche en sortie l'heure à Rio de Janeiro quand on donne en entrée l'heure à Paris.

<b>Variables</b>	$P$ et $R$ sont des flottants
<b>Entrée</b>	Demander la valeur de $P$
<b>Traitement</b>	Si $P \geq 5$ Alors Affecter à $R$ la valeur $P - 5$ . Sinon Affecter à $R$ la valeur $P + 19$ .
<b>Sortie</b>	Fin Si Afficher $R$

3. Programmer cet algorithme. [www](http://www).

Python	Scilab	Xcas
$P=\text{float}(\text{input}(P))$ $\text{if } P \geq 5:$ $\quad R=P-5$ $\text{else:}$ $\quad R=P+19$ $\text{print}(R)$	$P=\text{input}(P)$ $\text{if } P \geq 5:$ $\quad R=P-5$ $\text{else:}$ $\quad R=P+19$ $\text{afficher}(R)$	$\text{saisir}(P)$ $\text{si } P \geq 5$ $\quad alors R:=P-5$ $\quad sinon R:=P+19$ $\quad fsi$ $\quad afficher(R)$

**EXERCICE 25**

Analyser une situation • Écrire un programme

Un club sportif fait fabriquer des tee-shirts au nom du club. Chaque tee-shirt est facturé 4 €, mais ils sont facturés 3,50 € l'un si la commande est d'au moins 100 unités du produit.

Soit  $N$  la variable égale au nombre de tee-shirts commandés et  $P$  le prix payé par le club.

1. Calculer  $P$  dans les cas suivants : a.  $N = 40$  ;

b.  $N = 130$ .

a. Pour  $N = 40$ , on a :  $P = 40 \times 4$ , soit 160 €.

b. Pour  $N = 130$ , on a :  $P = 130 \times 3,5$ , soit 455 €.

2. Écrire ci-dessous le programme permettant d'obtenir le prix payé par le club en sortie pour un nombre  $N$  de tee-shirts commandés.

Python	Scilab	Xcas
$N=\text{int}(\text{input}(N))$ $\text{if } N \leq 100:$ $\quad P=N*4$ $\text{else:}$ $\quad P=N*3,5$ $\text{print}(P)$	$N=\text{input}(N)$ $\text{if } N \leq 100 \text{ then } P=N*4$ $\quad \text{else } P=N*3,5$ $\text{end}$ $\text{disp}(P)$	$saisir(N)$ $\text{si } N \leq 100$ $\quad alors P:=N*4$ $\quad sinon P:=N*3,5$ $\quad fsi$ $\quad afficher(P)$

**EXERCICE 26**

Comprendre un programme

On donne le programme suivant, écrit en langage Python. (or est l'écriture dans le langage Python du connecteur ou vu en raisonnement logique.)

1. Qu'obtient-on en sortie :

a. si on saisit « oui » en entrée ? On obtient : oui.

b. si on saisit « yes » en entrée ? On obtient : mauvaise réponse.

```
rep=input("répondre par oui ou non:")
if rep == "oui" or rep=="non":
    print(rep)
else:
    print("mauvaise réponse")
```

2. Décrire ce que fait ce programme. Ce programme demande de répondre par oui ou par non.

Si l'utilisateur répond correctement, la réponse s'affiche, sinon l'affichage est : mauvaise réponse.

**EXERCICE 27**

Moyenne

Compléter un algorithme • Modifier un algorithme

On considère l'algorithme ci-contre.

1. Compléter cet algorithme afin qu'il affiche en sortie si un candidat à un examen est reçu ou non, connaissant ses notes sur 20 représentées par les variables  $X, Y, Z, T$  aux épreuves. Les deux premières épreuves ont pour coefficient 2 et les deux suivantes ont pour coefficient 1.

2. Un candidat non reçu est admis à passer un oral de rattrapage si sa moyenne est au moins 8.Modifier cet algorithme dans le cadre ci-contre afin qu'il affiche de plus si un candidat est admis ou non à passer cet oral.

<b>Variables</b>	$X, Y, Z, T, M$ sont des flottants
<b>Entrée</b>	Saisir $X, Y, Z, T$
<b>Traitement et sortie</b>	$M$ prend la valeur $(2X+2Y+Z+T)/6$ . Si $M \geq 10$ Alors Afficher « le candidat est reçu ». Sinon Afficher « le candidat n'est pas reçu ». Fin Si
<b>Si <math>M \geq 10</math></b>	Alors Afficher « le candidat est reçu ».
<b>Sinon <math>M \geq 8</math></b>	Alors Afficher « le candidat est admis à l'oral ». Sinon Afficher « le candidat n'est pas admis à l'oral ». Fin Si
<b>Fin Si</b>	

<b>Si <math>M \geq 10</math></b>	Alors Afficher « le candidat est reçu ».
<b>Sinon <math>M \geq 8</math></b>	Alors Afficher « le candidat est admis à l'oral ». Sinon Afficher « le candidat n'est pas admis à l'oral ». Fin Si
<b>Fin Si</b>	

**EXERCICE 24**

Comprendre un programme • Analyser une situation

On donne le programme ci-contre, écrit en langage Python.

1. Qu'affiche ce programme en sortie si on saisit en entrée :

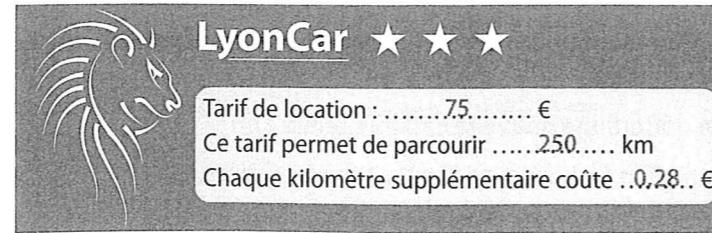
a.  $X = 162$  ? b.  $X = 625$  ?

a.  $162 \leq 250$ , donc  $C = 75$  ; il affiche 75.

b.  $625 > 250$ , donc  $C = 75 + 0,28 \times 375 = 180$  ; il affiche 180.

2. L'agence de location de voitures LyonCar utilise le programme ci-dessus afin de calculer le coût  $C$  de la location d'une voiture pour  $X$  kilomètres parcourus.

Compléter les informations manquantes dans la plaquette ci-contre diffusée par l'agence.



## Boucle bornée «POUR» (For)

### 1 Structure de la boucle «POUR»

On peut répéter un bloc d'instructions un certain nombre de fois fixé au départ : on utilise alors la boucle «POUR».

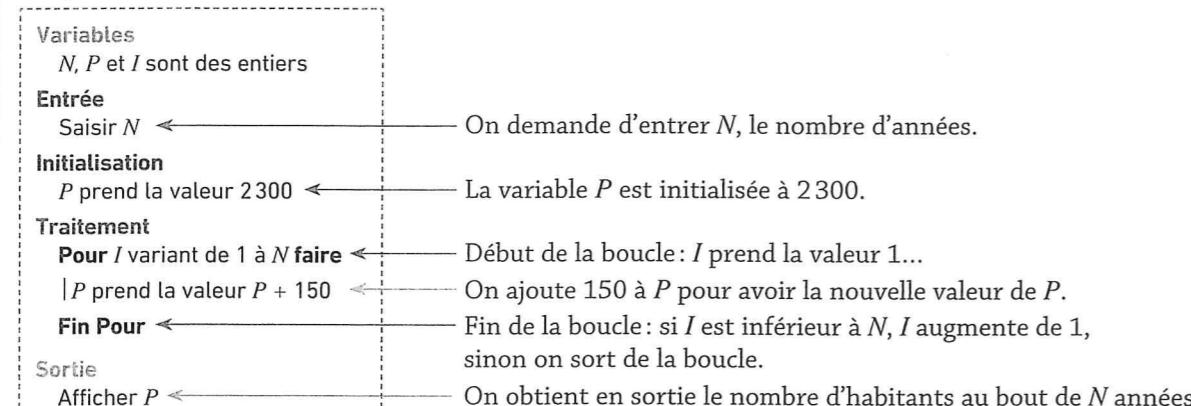
#### Exemple croquis d'un village

Un village compte aujourd'hui 2 300 habitants. Le village étant en pleine croissance, sa population augmente chaque année de 150 habitants.

On souhaite élaborer un algorithme donnant le nombre d'habitants de ce village dans  $N$  années.

Pour cela, on définit une variable  $P$  qu'on initialise à 2 300, puis on répète  $N$  fois l'opération qui consiste à ajouter 150 à  $P$  : le nombre de répétitions (ou «itérations») est connu au départ puisque c'est le nombre  $N$  d'années.

On parle de «calcul itératif».



On suppose qu'on a saisi  $N = 3$  en entrée.

	$I$	$P$
Avant le début de la boucle	2 300	← Au début, la variable $P$ est initialisée à 2 300.
Fin de la 1 <sup>re</sup> itération	1	2 450 ← $I = 1$ et $P$ prend la valeur $P + 150$ donc $P = 2 300 + 150 = 2 450$ .
Fin de la 2 <sup>e</sup> itération	2	2 600 ← $I = 2$ et $P$ prend la valeur $P + 150$ donc $P = 2 450 + 150 = 2 600$ .
Fin de la 3 <sup>e</sup> itération	3	2 750 ← $I = 3$ et $P$ prend la valeur $P + 150$ donc $P = 2 600 + 150 = 2 750$ .

La variable  $I$  contrôle le nombre de répétitions. Dans l'exemple ci-dessus,  $I$  prend pour valeur initiale 1 et pour valeur finale  $N$  en augmentant de 1 à chaque fois (cette augmentation constante est appelée «pas»).

### 2 Programmation

Programmation de la boucle bornée :

Python	Scilab	Xcas
<b>Pour <math>I</math> variant de <math>I_0</math> à <math>N</math> faire</b>   {instructions} <b>Fin Pour</b>	<b>for <math>I</math> in range(<math>I_0, N + 1</math>):</b> {instructions}	<b>for <math>I = I_0 : N</math></b> {instructions}

Programmes relatifs à l'algorithme vu en exemple :

Python	Scilab	Xcas
<pre>N=int(input("N=")) P=2300 for I in range(1,N+1):     P=P+150 print(P)</pre>	<pre>N=input("N=") P=2300 for I=1:N     P=P+150 end disp(P)</pre>	<pre>saisir(N); P:=2300; pour I de 1 jusque N faire     P:=P+150; fpour; afficher(P)</pre>

## EXERCICE 28

Comprendre un algorithme

1. Compléter le tableau ci-dessous et en déduire la valeur de  $A$  affichée en sortie de l'algorithme.

Variables	$A$ est un flottant
Entrée	$I$ est un entier
Initialisation	$A$ prend la valeur 100
Traitement	Pour $I$ variant de 1 à 4 faire   $A$ prend la valeur $A + 25$
Sortie	Fin Pour Afficher $A$

$I$	$A$
Avant le début de la boucle	100
Fin de la 1 <sup>re</sup> itération	125
Fin de la 2 <sup>e</sup> itération	150
Fin de la 3 <sup>e</sup> itération	175
Fin de la 4 <sup>e</sup> itération	200

La valeur de  $A$  affichée en sortie est : 200.

2. Si l'instruction «Afficher  $A$ » est placée entre les instructions « $A$  prend la valeur  $A + 25$ » et «Fin Pour», que permet d'afficher cet algorithme ainsi modifié ?

L'algorithme permet dans ce cas d'afficher les quatre valeurs : 125 ; 150 ; 175 ; 200.

## EXERCICE 29

Comprendre un programme

On considère le programme ci-dessous, écrit en langage Python.

1. Que permet d'obtenir ce programme ?

On obtient la phrase : Je dis : bravo bravo bravo bravo !

2. Comment modifier ce programme pour obtenir huit fois le mot «bravo» dans la phrase affichée ?

On peut remplacer l'instruction `range(1,5)` par `range(1,9)` puisque `i` prend alors successivement les valeurs entières de 1 à 8.

## EXERCICE 30

Moyenne

Compléter et programmer un algorithme

Mme Langlois effectue un unique versement de 20 000 euros sur un livret d'épargne.

Le capital est augmenté chaque année de 100 euros par le versement d'intérêts.

1. Déterminer le capital disponible au bout d'un an sur le livret d'épargne de Mme Langlois.

Le capital disponible au bout d'un an est 20 100 €.

2. Compléter l'algorithme ci-dessous pour qu'il affiche le capital disponible sur le livret au bout de 15 ans.

Variables	$C$ est un flottant
Entrée	$I$ est un entier
Initialisation	$C$ prend la valeur 20 000
Traitement	Pour $I$ variant de 1 à ... 15... faire   $C$ prend la valeur $C + 100$
Sortie	Afficher $C$

3. Programmer cet algorithme, puis déterminer le capital disponible au bout de 15 ans. [www.](http://www.)

Le capital disponible au bout de 15 ans est 21 500 €.

$C=20000$   
for  $I$  in range(1,16):  
..... $C=C+100$   
print( $C$ )

**EXERCICE 31**

## Compléter et programmer un algorithme

En prévision d'une course de vélo, Fanny suit le programme d'entraînement suivant sur douze samedis : elle parcourt 25 kilomètres le premier samedi, puis augmente chaque semaine de 11 kilomètres la distance parcourue.

- Déterminer la distance  $D$  parcourue le deuxième samedi et la distance totale  $T$  parcourue au bout de deux samedis d'entraînement.

$$D = 25 + 11 = 36. \text{ Le deuxième samedi, Fanny parcourt 36 kilomètres.}$$

Le premier samedi, Fanny parcourt 25 kilomètres, le deuxième samedi elle parcourt 36 kilomètres, donc la distance totale parcourue au bout de deux samedis est  $T = 25 + 36$ , soit 61 kilomètres.

- Compléter l'algorithme ci-dessous pour qu'il affiche la distance totale parcourue  $T$  au bout des douze samedis d'entraînement.

<b>Variables</b>	$D$ et $T$ sont des flottants
	$I$ est un entier
<b>Initialisation</b>	$D$ prend la valeur 25
	$T$ prend la valeur 25
<b>Traitement</b>	Pour $I$ variant de 2 à 12 faire
	$D$ prend la valeur $D + 11$
	$T$ prend la valeur $T + D$
	Fin Pour
<b>Sortie</b>	Afficher $T$

- Programmer cet algorithme et déterminer la distance totale parcourue à la fin des douze samedis d'entraînement.

```
D=25
T=25
for I in range(2,13):
    D=D+11
    T=T+D
print(T)
```

La distance totale parcourue à la fin des douze samedis d'entraînement est 1 026 kilomètres.

**EXERCICE 32**

## Analyser une situation et compléter un programme

L'an prochain, Oscar consacrera 1 800 € pour ses loisirs, puis pour faire des économies, il prévoit à partir de l'année suivante de réduire chaque année de 5 % les dépenses de ce secteur.

- Calculer le budget loisir  $B$  d'Oscar dans deux ans.

Le budget loisir  $B$  d'Oscar dans deux ans sera de 1 710 €.

- Quel budget total aura-t-il consacré à ses loisirs au cours des deux prochaines années ?

Ce budget total sera de :  $1\,800 + 1\,710 = 3\,510$  €.

- Compléter le programme ci-dessous écrit en langage Python afin qu'il affiche le budget total  $T$  qu'Oscar aura consacré à ses loisirs au cours des dix prochaines années s'il respecte son plan d'économies.

```
B=1800
T=0
for I in range (1,...,11):
    I=I+B
    B=0.95*B
print(T)
```

© Bordas 2017 - La photocopie non autorisée est un délit.

**EXERCICE 33**

## Comprendre et programmer un algorithme

Soit  $n$  un entier naturel non nul. On pose  $S = 1 + 2 + \dots + n$ .

- Calculer les sommes  $1 + 2$  ;  $1 + 2 + 3$  et  $1 + 2 + 3 + 4$ , puis calculer  $S$  lorsque  $n = 5$ .

$$1+2=3; 1+2+3=6; 1+2+3+4=10; 1+2+3+4+5=15.$$

- On considère l'algorithme ci-dessous.

<b>Variables</b>	$S$ , $n$ et $I$ sont des entiers
<b>Entrée</b>	Saisir $n$
<b>Initialisation</b>	$S$ prend la valeur 0
<b>Traitement</b>	Pour $I$ variant de 1 à $n$ faire $S$ prend la valeur $S + I$
<b>Fin Pour</b>	
<b>Sortie</b>	Afficher $S$

À quoi correspond la valeur affichée en sortie d'algorithme ?

La valeur affichée est celle de la somme  $1 + 2 + 3 + 4 + \dots + n$ .

- Programmer cet algorithme et en déduire la somme des 1 000 premiers entiers non nuls. [www.](#)

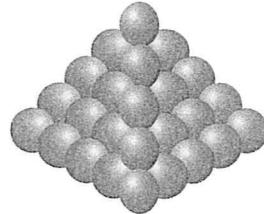
```
n=int(input("n="))
S=0
for I in range(1,n+1):
    S=S+I
print(S)
```

La somme des 1 000 premiers entiers non nuls est :

$$1+2+3+\dots+1\,000=500\,500.$$

**EXERCICE 34**

## Analyser une situation et écrire un programme



On empile des sphères, formant ainsi une pyramide de base carrée.

- Déterminer le nombre de sphères nécessaires pour constituer une pyramide de deux « niveaux ».

Il faut 5 sphères : 4 au 1<sup>er</sup> niveau et une au 2<sup>nd</sup> niveau.

- Déterminer le nombre de sphères nécessaires pour constituer une pyramide de cinq « niveaux ».

Il faut 55 sphères : une au 5<sup>th</sup> niveau, 4 au 4<sup>th</sup>, 9 au 3<sup>rd</sup>, 16 au 2<sup>nd</sup> et 25 au 1<sup>st</sup>.

- Sachant qu'il faut 140 sphères pour construire une pyramide de sept « niveaux », déterminer le nombre de sphères nécessaires pour construire une pyramide de huit « niveaux ».

$$140 + 8^2 = 204, \text{ donc il faut 204 sphères.}$$

- En utilisant la structure ci-dessous, écrire un programme en langage Python permettant de déterminer le nombre  $B$  de sphères nécessaires pour constituer une pyramide comportant  $N$  « niveaux » à partir de la saisie de l'entier  $N$ .

```
N=int(input("N="))
B=0
for I in range (1,N+1):
    B=B+I**2
print(B)
```

# F Boucle non bornée « TANT QUE » (While)

## 1 Structure de la boucle « TANT QUE »

On peut répéter un bloc d'instructions tant qu'une condition reste vérifiée : on utilise alors la boucle TANT QUE.

### Exemple : les rebonds d'une balle

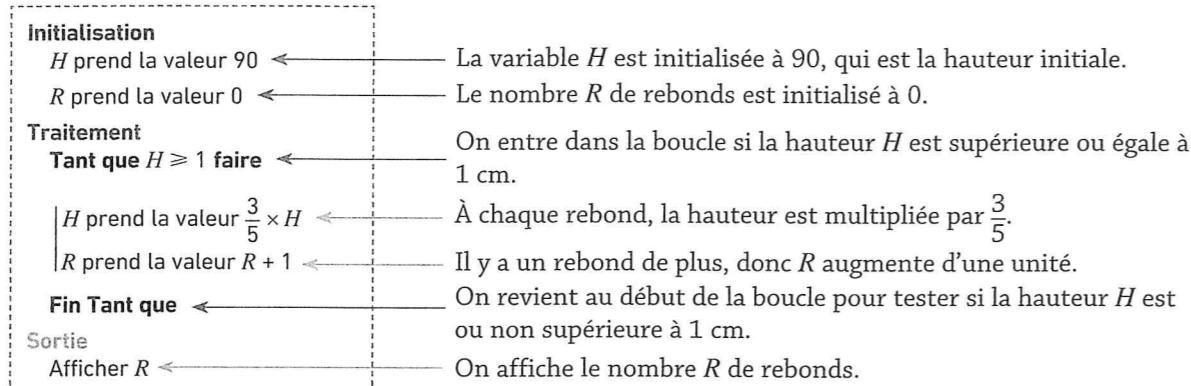
Une balle lâchée d'une hauteur de 90 centimètres rebondit chaque fois qu'elle touche le sol au  $\frac{3}{5}$ <sup>e</sup> de la hauteur du rebond précédent.

On souhaite élaborer un algorithme donnant le nombre de rebonds au bout duquel la hauteur atteinte par la balle est strictement inférieure à 1 centimètre.

On définit la variable  $H$  égale à la hauteur du rebond exprimée en centimètres. On doit répéter plusieurs fois l'instruction «  $H$  prend la valeur  $\frac{3}{5} \times H$  », sans savoir à l'avance le nombre de ces répétitions.

Pour cela, on teste si la variable  $H$  est supérieure ou égale à 1 en début de boucle et le traitement dans la boucle est réalisé tant que cette condition est vérifiée.

Pour compter le nombre de rebonds effectués par la balle, on introduit un « compteur »  $R$  : il est initialisé à 0, et, chaque fois que la boucle est parcourue, sa valeur augmente d'une unité.



C'est la boucle « Tant que... » qui permet de répéter ce calcul. Cette boucle est utilisée lorsque l'on veut recommencer un même bloc d'instructions jusqu'à valider une condition de sortie donnée à l'avance.

Avec une hauteur initiale de 90 cm, les valeurs des variables  $R$  et  $H$  (en centimètres) sont données ci-contre.

C'est la structure itérative avec fin de boucle conditionnelle.

Étapes	$H$	$R$	Condition vérifiée
Avant le début de la boucle	90	0	Oui
1 <sup>er</sup> passage dans la boucle	54	1	Oui
2 <sup>e</sup> passage dans la boucle	32,4	2	Oui
...	...	...	...

## 2 Programmation

	Python	Scilab	Xcas
Tant que {condition C} faire   {instructions} Fin Tant que	while {condition C} : {instructions} end	while {condition C} {instructions} end	tantque {condition C} faire {instructions} ftantque

Programmes relatifs à l'algorithme vu en exemple :

Python	Scilab	Xcas
<pre>H=90 R=0 while H≥1:     H=H*3/5     R=R+1 print(R)</pre>	<pre>H=90 R=0 while H≥1     H=H*3/5     R=R+1 end disp(R)</pre>	<pre>H:=90; R:=0; tantque H≥1 faire     H:=H*3/5;     R:=R+1 ftantque; afficher (R)</pre>

On trouve  $R = 9$ . C'est au 9<sup>e</sup> rebond que la hauteur atteinte par la balle est strictement inférieure à 1 cm.

## EXERCICE 35

Comprendre un algorithme

On considère l'algorithme ci-contre.

1. Quelle condition doit être remplie pour que la boucle « Tant que » soit réalisée ?

On doit avoir  $A > 9$ .

2. On saisit 5 en entrée. Quelle valeur est affichée en sortie ?

5 est affiché en sortie.

3. Peut-on obtenir 21 en sortie ? Justifier.

Non, on ne peut pas parce que si  $A$  contient la valeur 21, on ne sort pas de la boucle « Tant que ».

4. On saisit 30 en entrée. Compléter le tableau suivant et en déduire la valeur affichée en sortie d'algorithme.

Étapes	$A$	Condition vérifiée ?
Avant la boucle	30	oui
1 <sup>er</sup> passage dans la boucle	21	oui
2 <sup>e</sup> passage dans la boucle	12	oui
3 <sup>e</sup> passage dans la boucle	3	non

La valeur affichée en fin d'algorithme est : 3.

## EXERCICE 36

Comprendre un algorithme

En 2014, Carole verse sur son livret d'épargne 3 000 €.

Chaque année, la somme disponible sur le livret est multipliée par 1,03. L'algorithme ci-contre permet d'afficher l'année à partir de laquelle Carole disposera pour la première fois d'au moins 3 500 €.

1. Quelle variable contient les valeurs successives de l'épargne disponible ?

C'est la variable  $S$  qui contient les valeurs successives de l'épargne disponible.

2. Quel est le rôle de la variable  $A$  ?

La variable  $A$  contient les valeurs successives des années, depuis 2014 jusqu'à l'année solution.

3. Compléter le tableau suivant et en déduire la valeur affichée en sortie.

Étapes	$S$	$A$	Condition vérifiée ?
Avant la boucle	3 000	2014	oui
1 <sup>er</sup> passage dans la boucle	3 090	2015	oui
2 <sup>e</sup> passage dans la boucle	3 182,7	2016	oui
3 <sup>e</sup> passage dans la boucle	3 278,18	2017	oui
4 <sup>e</sup> passage dans la boucle	3 376,53	2018	oui
5 <sup>e</sup> passage dans la boucle	3 477,82	2019	oui
6 <sup>e</sup> passage dans la boucle	3 582,16	2020	non

La valeur affichée en sortie est : 2020.

Variables	$S$ est un flottant
Initialisation	$A$ est un entier
Traitement	$S$ prend la valeur 3 000
	$A$ prend la valeur 2014
	Tant que $S < 3 500$ faire
	Affecter à $S$ la valeur $S \times 1,03$
	Affecter à $A$ la valeur $A + 1$
	Fin Tant que
Sortie	Afficher $A$

**EXERCICE 37**

Compléter et programmer un algorithme

Dans une petite ville, on constate depuis quelques années une hausse annuelle de 5 % du prix des loyers. Cette année, le prix moyen de location du mètre carré est 8 €.

1. Quel sera le prix de location du mètre carré dans un an ?

$$8 \times \left(1 + \frac{5}{100}\right) = 8 \times 1,05 = 8,4. \text{ Dans un an, le prix de location du mètre carré sera de } 8,4 \text{ €.}$$

2. On souhaite déterminer dans combien d'années le prix moyen de location du mètre carré dépassera 12 €.

On veut compléter l'algorithme ci-contre afin qu'il affiche le nombre  $N$  d'années cherché.

a. Compléter l'algorithme.

b. Programmer cet algorithme, puis répondre à la question posée. [www.](#)

La valeur affichée est 9.

Dans 9 ans, le prix moyen de location du mètre carré dépassera pour la première fois 12 €.

<b>Variables</b>	$L$ est un flottant $N$ est un entier
<b>Initialisation</b>	$L$ prend la valeur 8 $N$ prend la valeur 0
<b>Traitement</b>	Tant que $L \leq 12$ faire $L$ prend la valeur $1,05 \times L$ $N$ prend la valeur $N + 1$ Fin Tant que
<b>Sortie</b>	Afficher $N$

**EXERCICE 38**

Comprendre un algorithme

Le 1<sup>er</sup> janvier 2015, David a reçu 90 € d'étrennes, puis chaque année, celles-ci augmentent de 5 €. David décide de ne pas dépenser cet argent avant de disposer de 700 €.



1. Déterminer le montant  $E$  des étrennes que David reçoit le 1<sup>er</sup> janvier 2016.

David reçoit 95 €.

2. En déduire la somme totale  $T$  dont il dispose en 2016.

$$T = 90 + 95 = 185. \text{ En 2016, il dispose de } 185 \text{ €.}$$

3. L'algorithme ci-contre permet d'afficher l'année à partir de laquelle David disposera d'au moins 700 €.

a. Faire fonctionner cet algorithme en complétant le tableau ci-dessous.

b. En déduire à partir de quelle année, David pourra dépenser son argent.

L'année cherchée est 2021.

<b>Variables</b>	$E, T$ et $A$ sont des entiers
<b>Initialisation</b>	$E$ prend la valeur 90 $T$ prend la valeur 90 $A$ prend la valeur 2015
<b>Traitement</b>	Tant que $T < 700$ faire Affecter à $E$ la valeur $E + 5$ Affecter à $T$ la valeur $T + E$ Affecter à $A$ la valeur $A + 1$ Fin Tant que
<b>Sortie</b>	Afficher $A$

Étapes	$E$	$T$	$A$	Condition vérifiée ?
Avant la boucle	90	90	2015	oui
1 <sup>er</sup> passage dans la boucle	95	185	2016	oui
2 <sup>e</sup> passage dans la boucle	100	285	2017	oui
3 <sup>e</sup> passage dans la boucle	105	390	2018	oui
4 <sup>e</sup> passage dans la boucle	110	500	2019	oui
5 <sup>e</sup> passage dans la boucle	115	615	2020	oui
6 <sup>e</sup> passage dans la boucle	120	735	2021	non

**EXERCICE 39**

Analyser une situation et écrire un algorithme

Une entreprise de forage creuse des puits dans le désert afin d'atteindre la nappe d'eau phréatique. Cette entreprise facture le premier mètre creusé 100 €, le second mètre 140 € et ainsi de suite en augmentant le prix de chaque nouveau mètre creusé de 40 €.

1. Calculer le prix  $M$  du troisième mètre creusé, puis le prix total  $S$  d'un puits de trois mètres de profondeur.

$$\text{Prix du troisième mètre : } M = 180 \text{ €. On a } S = 100 + 140 + 180 = 420.$$

2. a. Compléter le programme ci-contre, écrit en langage Python, afin qu'il affiche le prix total  $S$  d'un puits de  $H$  mètres de profondeur lorsque l'on entre la valeur de  $H$ .

b. Écrire ce programme puis l'exécuter afin de déterminer le prix d'un puits de 8 mètres de profondeur, puis celui d'un puits de 12 mètres de profondeur. [www.](#)

Un puits de 8 mètres de profondeur coûte 1 920 €.

Un puits de 12 mètres de profondeur coûte 3 840 €.

3. Une organisation humanitaire dispose d'un budget de 4 000 €.

a. En utilisant le programme de la question 2., déterminer la profondeur maximale d'un puits que peut financer l'organisation.

Pour  $H = 12$ , le prix est de 3 840 €.

Pour  $H = 13$ , le prix est de 4 420 €.

La profondeur maximale est de 12 mètres.

b. Compléter l'algorithme ci-contre afin qu'il affiche cette profondeur maximale.

```
M=100
S=100
N=1
H=int(input("H="))
while N<H:
    M=M+40
    S=S+M
    N=N+1
print(S)
```

<b>Variables</b>	$M, S$ et $N$ sont des entiers
<b>Initialisation</b>	$M$ prend la valeur 100
	$S$ prend la valeur 100
	$N$ prend la valeur 0
<b>Traitement</b>	Tant que $S \leq 4000$ faire
	$M$ prend la valeur $M + 40$
	$S$ prend la valeur $S + M$
	$N$ prend la valeur $N + 1$
<b>Sortie</b>	Fin Tant que
	Afficher $N$

**EXERCICE 40**

Comprendre un programme

En 2016, les rejets polluants d'un groupe industriel sont évalués à 5 000 tonnes.

Le groupe est contraint de réduire ses rejets polluants de 8 % chaque année jusqu'à ce que ceux-ci ne dépassent pas 2 000 tonnes annuelles. On suppose que le groupe respecte ce plan de réduction.

1. Par quelle valeur est multipliée chaque année la quantité de rejets polluants ?

La quantité de rejets polluants est multipliée chaque année par 0,92.

2. Dans le programme ci-contre, écrit en langage Python, la variable  $R$  est la quantité annuelle de polluant rejeté.

a. Quel rôle joue la variable  $T$  ?

$T$  contient la quantité cumulée de polluants rejetés depuis

le début de l'année 2016.

b. Compléter ce programme.

c. Les valeurs affichées sont 2027 et environ 39 521. Interpréter ces deux valeurs.

En 2027, la quantité annuelle de polluants rejetés sera pour la première

fois inférieure à 2 000 tonnes. La quantité totale de polluants rejetés, depuis le début de l'année 2016 jusqu'à la fin de l'année 2027, aura été d'environ 39 521 tonnes.

```
R=5000
T=5000
N=2016
while R>2000:
    R=R*0,92
    T=T+R
    N=N+1
print(N)
print(T)
```

## C Fonctions

On peut simplifier l'écriture des programmes en utilisant des **fonctions**, sur le modèle des fonctions numériques étudiées en mathématiques.

### 1 Structure d'une fonction

#### Exemple 1 : calcul de la surface de la peau

La surface  $S$  en mètres carrés de la peau d'un adulte est donnée approximativement par la formule :  

$$S = \frac{\sqrt{L \times M}}{6}$$
, où  $L$  est la taille de la personne exprimée en mètres et  $M$  sa masse exprimée en kilogrammes.

Pour programmer ce calcul, on peut définir une fonction appelée **surf** qui aura pour paramètres  $L$  et  $M$  :

**fonction surf( $L,M$ )**

$$\text{returne } \left( \frac{\sqrt{L \times M}}{6} \right)$$

au lieu de définir les entrées  $L$  et  $M$  comme d'habitude, ces variables sont des **arguments** de la fonction et celle-ci sera écrite **surf( $L,M$ )** dans le programme.

Le résultat est obtenu par l'instruction **returne** (ou **return**).

On obtient la surface corporelle d'un adulte de 1,80 m et 75 kg avec **surf(1.8,75)** : on trouve environ 1,94 m<sup>2</sup>.

#### Exemple 2 : détermination du plus grand de deux nombres

On peut définir une fonction qui détermine le plus grand de deux nombres  $a$  et  $b$  donnés. Les arguments de cette fonction **maxi** sont  $a$  et  $b$ , et le résultat renvoyé par l'instruction **returne** est le plus grand de ces deux nombres.

```
fonction maxi(a,b)
  Si a≥b
    Alors returne (a)
    Sinon returne (b)
  Fin Si
```

Une fonction est un bloc d'instructions qui a reçu un nom, dont le fonctionnement dépend d'un certain nombre de paramètres (les **arguments** de la fonction) et qui renvoie un résultat (au moyen de l'instruction **returne**).

**Remarques :**

- 1) Les entrées d'une fonction se font en donnant des valeurs aux arguments.
- 2) Une fonction peut n'avoir aucun argument.
- 3) On peut aussi utiliser l'instruction **print** dans une fonction. Contrairement à l'instruction **returne**, le programme continue après **print** alors qu'il s'arrête après avoir rencontré **returne**.
- 4) Une fonction peut être appelée dans un programme ou dans une autre fonction. Ainsi, les fonctions permettent d'éviter des redondances et de structurer les programmes.

### 2 Programmation

	Python	Scilab	Xcas
	<pre>fonction mafonc(a, b...) {instructions} returne(résultat)</pre>	<pre>function y=mafonc(a, b...) y=..... endfunction</pre>	<pre>mafonc := {..... returne c}</pre>

Programmes relatifs aux algorithmes vus en exemple :

	Python	Scilab	Xcas
surf	<pre>def surf(L,M):     return(sqrt(L*M)/6)</pre>	<pre>function y=surf(L, M)     y=sqr(L*M)/6 endfunction</pre>	<pre>surf(L,M) := returne sqrt(L*M)/6</pre>
maxi	<pre>def maxi(a,b):     if a&gt;=b:         return(a)     else:         return(b)</pre>	<pre>function y=maxi(a, b);     if a&gt;=b then y=a;     else y=b;     end endfunction</pre>	<pre>maxi (a, b) := si a&gt;=b alors returne a sinon returne b fsi</pre>

## EXERCICE 41 Fonctions affines

Comprendre une fonction

1. Soit  $f$  la fonction programmée ci-contre en langage Python.

a. Que renvoie  $f(1)$  ?  $f(-2)$  ?

$f(1)$  renvoie  $3 \times 1 - 1 = 2$ .  $f(-2)$  renvoie  $3 \times (-2) - 1 = -7$ .

b. Que permet de faire cette fonction ?

Cette fonction permet de calculer l'image d'un réel  $x$  par la fonction numérique qui à  $x$  associe  $3x - 1$ .

2. On a programmé une nouvelle fonction nommée **affine**.

a. Quels sont les arguments de cette fonction ?

Cette fonction a trois arguments :  $a$ ,  $b$  et  $x$ .

b. Quel va être l'affichage ?

$\text{affine}(1,2,3)$  renvoie  $1 \times 3 + 2 = 5$ ; l'affichage sera 5.

c. Comment peut-on afficher avec cette fonction l'image du nombre 4 par la fonction numérique qui à  $x$  associe  $-5x + 2$  ?

On peut, soit demander  $\text{affine}(-5,2,4)$  dans la console, soit utiliser l'instruction `print(affine(-5,2,4))`.

```
def f(x):  
    return(3*x-1)
```

```
def affine(a,b,x):  
    return(a*x+b)  
  
print(affine(1,2,3))
```

## EXERCICE 42 Programmer une fonction

Bainbridge a proposé une formule donnant la vitesse de nage  $v$  d'un poisson (en cm.s<sup>-1</sup>) en fonction de sa longueur  $L$  (exprimée en cm) et de la fréquence  $f$  des battements de sa queue par seconde :  $v = \frac{1}{4} L (3f - 4)$ .

1. Calculer la vitesse d'un poisson de 16 cm et qui avance avec 15 battements de queue par seconde.

$V = \frac{1}{4} \times 16 \times (3 \times 15 - 4)$ , soit  $V = 164$  cm.s<sup>-1</sup>.

2. Programmer une fonction permettant d'obtenir la vitesse de nage d'un poisson connaissant sa longueur et la fréquence de ses battements de queue.

Python	Scilab	Xcas
<pre>def vitesse(L,f):     ....     return(L*(3*f-4)/4)</pre>	<pre>function y=vitesse(L,f)... .... y=L*(3*f-4)/4 ....</pre>	<pre>vitesse(L,f):=..... {returne ..L*(3*f-4)/4}; ....endfunction</pre>

3. Comment peut-on afficher avec cette fonction la vitesse d'un poisson de longueur 30 cm et dont la fréquence des battements de queue est de 20 battements par seconde ?

On peut, soit demander  $\text{vitesse}(30,20)$  dans la console, soit utiliser l'instruction `print(vitesse(30,20))`.

## EXERCICE 43 Exécuter une fonction

1. Soit la fonction **comp** programmée ci-contre dans le langage Python.

Qu'obtient-on comme affichage après avoir appelé cette fonction ?

Pour  $i = 1$ , `print(i)` affiche 1 ; puis, pour  $i = 2$ , `print(i)` affiche 2, et ainsi de suite...

jusqu'à  $i = 9$ . On obtient donc l'affichage : 1. 2. 3. 4. 5. 6. 7. 8. 9.

2. On modifie la fonction **comp**. Qu'obtient-on comme affichage après avoir appelé la fonction **pair** ?

On obtient l'affichage : 2. 4. 6. 8. 10. 12. 14. 16. 18. 20.

```
def comp():  
    i=1  
    while i<10:  
        print(i)  
        i=i+1
```

```
def pair():  
    i=1  
    while i<=10:  
        k=2*i  
        print(k)  
        i=i+1
```

**EXERCICE 44**

## Compléter une fonction • Créer une fonction

1. Compléter la fonction `mini`, dont on donne le programme incomplet ci-dessous en langage Python, afin qu'elle retourne en sortie le plus petit des deux nombres.

2. Programmer une fonction `mini4` permettant de déterminer le plus petit de quatre nombres donnés, en faisant appel à la fonction `mini` définie dans la première question.

```
def mini(a,b):
    if(a < b):
        return(a)
    else:
        return(b)

def mini4(a,b,c,d):
    e=mini(a,b)
    f=mini(c,d)
    return(mini(e,f))
```

**EXERCICE 45**

## Compléter un programme

Compléter le programme ci-dessous, écrit en langage Python, afin qu'il affiche le cube des dix premiers entiers naturels non nuls.

```
def cube(x):
    return(x*x*x)

for k in range(1,11):
    print(cube(k))
```

**EXERCICE 46**

## Exécuter et comprendre une fonction

La fonction `fac`, dont l'argument `a` est un entier naturel non nul, est programmée ci-dessous en langage Python.

1. a. Que renvoie `fac(3)` ?

Puisque  $a+1=4$ , l'entier  $i$  varie de 1 à 3. Le nombre  $b$  vaut 1 au départ, puis  $1 \times 1 = 1$ , puis  $1 \times 2 = 2$ , puis  $2 \times 3 = 6$ . Ainsi, `fac(3)` renvoie 6.

- b. Que renvoie `fac(6)` ?

Après la valeur initiale 1,  $b$  prend successivement les valeurs 1, 2, 6, 24, 120 et 720. Donc `fac(6)` renvoie 720.

2. Comment peut-on calculer le produit  $1 \times 2 \times 3 \times 4 \times 5$  en utilisant la fonction `fac` ?

On saisit `fac(5)` dans la console.

3. Que calcule cette fonction pour un entier `a` non nul ?

Pour un entier `a` non nul, cette fonction calcule le produit des entiers 1, 2, 3... jusqu'à `a`.

```
def fac(a):
    b=1
    for i in range(1,a+1):
        b=b*i
    return(b)
```

**EXERCICE 47**

## Chaînes de caractères

## Compléter et modifier une fonction

On veut programmer en langage Python une fonction qui a pour argument un mot ou une phrase et qui renvoie en sortie le nombre de fois où la lettre « e » se trouve dans ce mot ou cette phrase. Dans la fonction ci-dessous, `len(ph)` calcule la longueur de la chaîne de caractères `ph`.

1. Compléter le programme ci-dessous.

- 2.Modifier cette fonction afin qu'elle renvoie la fréquence de la lettre « a » dans la phrase ou le mot.

On change la ligne « `ph[i]==="e"` » en « `ph[i]==="a"` ». Avant la dernière ligne, on introduit la ligne : `f=n/s`, puis on change l'instruction `return(n)` en `return(f)`.

```
def edansphrase(ph):
    s=len(ph)
    n=0
    for i in range(0,s):
        if ph[i]==="e":
            n=n+1
    return(n)
```

**EXERCICE 48**

## Triangle rectangle

## Exécuter et comprendre une fonction • Compléter une fonction

1. On a programmé une fonction `f` dans le langage Python.

a. Que renvoie `f(3,4,5)` ? `f(3,4,5).renvoie.0.`

b. Que renvoie `f(5,4,3)` ? `f(5,4,3).renvoie.32.`

c. Que renvoie `f(a,b,c)` lorsque  $a, b, c$  sont les longueurs des trois côtés d'un triangle rectangle ?

`f(a,b,c).renvoie.0` seulement dans le cas où  $c$  représente la longueur du plus grand côté du triangle.

2. Compléter le programme de la fonction `trirec`, en faisant appel à la fonction `f`, afin que la fonction `trirec` renvoie si le triangle ABC de côtés  $a, b, c$  est ou non un triangle rectangle.

(*or est l'écriture dans le langage Python du connecteur ou vu en raisonnement logique.*)

```
def f(a,b,c):
    return(a**2+b**2-c**2)
```

```
def trirec(a,b,c):
    if ...f(a,b,c)==0... or ...f(a,c,b)==0... or ...f(b,c,a)==0...:
        return("triangle rectangle")
    else:
        return("triangle non rectangle")
```

**EXERCICE 49**

## Diviseurs d'un entier

## Exécuter et comprendre une fonction • Modifier une fonction

1. On a programmé en langage Python la fonction `nbdv` dont l'argument est un entier naturel `a` non nul. (*reste(a,i) donne le reste de la division de `a` par `i`.*)

- a. Que signifie l'instruction `reste(a,i)==0` ?

`reste(a,i)==0` teste si l'entier `a` est divisible par `i`.

- b. Que renvoie `nbdv(6)` ?

Puisque  $a+1=7$ , l'entier  $i$  varie de 1 à 6. Pour  $i=1$ , le test `reste(a,i)==0` est vérifié, donc `n` prend la valeur 1.

Puis le test est vérifié quand  $i$  vaut 2, 3 et 6. Donc, `nbdv(6)` renvoie 4.

- c. Que renvoie cette fonction pour un entier naturel `a` non nul ?

Cette fonction renvoie le nombre de diviseurs (positifs) de `a`.

```
def nbdv(a):
    n=0
    for i in range(1,a+1):
        if reste(a,i)==0:
            n=n+1
    return(n)
```

2. Modifier cette fonction en une fonction `divi` qui affiche tous les diviseurs d'un entier naturel `a` non nul.

Python	Scilab	Xcas
<pre>def divi(a):     for i in range(1,a+1):         if reste(a,i)==0:             print(i)</pre>	<pre>function y=divi(a)     for i=1:a         if reste(a,i)==0 then disp(i);     end endfunction</pre>	<pre>divi(a):={...} pour k de 1...jusque a faire     si irem(a,k)==0 alors afficher(k) fsi; fpour;</pre>

**EXERCICE 50**

## Calculs de volumes

## Comprendre et compléter une fonction

On rappelle que le volume d'une pyramide est donné par la formule  $V = \frac{1}{3} \text{base} \times \text{hauteur}$ .

On a programmé ci-dessous, en langage Python, une fonction appelée `volpyr`.

```
def volpyr(base, hauteur):
    V=base*hauteur/3
    return(V)
```

1. Que renvoie cette fonction ?

Cette fonction renvoie le volume d'une pyramide dont on connaît l'aire de la base et la hauteur.

2. Compléter, en faisant appel à la fonction `volpyr`, la fonction programmée ci-dessous afin qu'elle renvoie le volume d'une pyramide de base rectangulaire à partir de sa longueur, sa largeur et sa hauteur.

```
def volpyrect(longueur, largeur, hauteur):
    V=volpyr(longueur*largeur, hauteur)
    return(V)
```

## A Généralités sur les fonctions – Variations

### EXERCICE 51

Intervalles

Instruction conditionnelle (If) → p. 14

1. Compléter l'algorithme suivant :

Variables	$A$ et $X$ sont des flottants
Entrée	Saisir $A$ et $X$
Traitement et sortie	<p>Si <math>X \geq A</math>...</p> <p>Alors Afficher « <math>X \in [A ; +\infty[</math> »</p> <p>Sinon Afficher « <math>X \notin [A ; +\infty[</math> »</p>
	Fin Si

2. Modifier l'algorithme précédent pour qu'il affiche en sortie si le nombre  $X$  saisi appartient ou non à l'intervalle  $[A ; B]$ , les réels  $A$  et  $B$  étant entrés en début d'algorithme.

Variables	$A, B$ et $X$ sont des flottants
Entrée	Saisir $A, B$ et $X$
Traitement et sortie	<p>Si <math>X \geq A</math> et <math>X \leq B</math>...</p> <p>Alors Afficher <math>X \in [A ; B]</math></p> <p>Sinon Afficher <math>X \notin [A ; B]</math></p>
	Fin Si

### EXERCICE 52

Intervalles

Instruction conditionnelle (If) → p. 14

1. Compléter le programme suivant écrit en langage Python.

```
X=float(input("X="))
if X>....2.... and X<....7....:
    print("Le nombre X appartient à l'intervalle ]2;7[")
```

2. En utilisant les connecteurs logiques and et or, écrire un programme qui affiche un message si le nombre  $X$  saisi, appartient à la réunion des intervalles  $[1 ; 9]$  et  $[14 ; 20]$ .

```
X=float(input("X="))
if...(X>=1 and X<=9)... or ...(X>=14 and X<=20)...
    print("Le nombre X appartient à la réunion des intervalles [1;9] et [14;20]...")
```

### EXERCICE 53

Fonctions : généralités

Algorithmes – Variables – Affectation → p. 3

On considère l'algorithme ci-contre.

1. On entre la valeur 2. Quelle est la valeur affichée en sortie ?

La valeur affichée en sortie est 6.

2. On entre la valeur -1. Quelle est la valeur affichée en sortie ?

La valeur affichée en sortie est 0.

3. Cet algorithme permet d'afficher l'image d'un réel  $X$  par une fonction  $f$ . Déterminer l'expression de  $f$ . $f(x)=x^3-x$ .

Variables	$X$ et $Y$ sont des flottants
Entrée	Demander la valeur de $X$
Traitement et sortie	Affecter à $Y$ la valeur $X^3 - X$
Sortie	Afficher $Y$

### EXERCICE 54

Fonctions : généralités

Boucle bornée « POUR » (For) → p. 18

Comprendre un programme

On considère le programme ci-contre écrit en langage Python.

1. On entre la valeur 3. Quelle est la valeur affichée en sortie ?

La valeur affichée est 27.

2. Ce programme permet d'afficher l'image d'un réel  $X$  par une fonction  $f$ . Déterminer l'expression de  $f$ . $f(x)=x^3$ .

```
X=float(input("X="))
Y=1
for I in range(1,4):
    Y=X*Y
print(Y)
```

### EXERCICE 55

Fonctions : ensemble de définition

Instruction conditionnelle (If) → p. 14

Compléter un algorithme • Modifier un algorithme

On considère la fonction  $g$  définie sur  $] -1 ; +\infty[$  par  $g(x) = \frac{2+x^2}{x+1}$ .1. Compléter l'algorithme ci-contre pour qu'il affiche en sortie l'image par la fonction  $g$  du réel  $X$  entré.

Variables	$X$ et $Y$ sont des flottants
Entrée	Saisir la valeur de $X$
Traitement	$Y$ prend la valeur $\frac{2+x^2}{x+1}$
Sortie	Afficher $Y$

2. Modifier l'algorithme précédent pour qu'il affiche un message d'erreur si le nombre saisi n'appartient pas à l'ensemble de définition de  $g$  et qu'il calcule son image s'il appartient à l'ensemble de définition.

Variables	$X$ et $Y$ sont des flottants
Entrée	Saisir la valeur de $X$
Traitement et sortie	<p>Si <math>X \leq -1</math>...</p> <p>Alors Afficher « Erreur : <math>X \notin ]-1 ; +\infty[</math> »</p> <p>Sinon <math>Y</math> prend la valeur <math>\frac{2+x^2}{x+1}</math></p> <p>Afficher <math>Y</math></p>
	Fin Si

### EXERCICE 56

Fonctions : courbe représentative

Instruction conditionnelle (If) → p. 14

Compléter un algorithme • Programmer un algorithme

On considère la fonction  $f$  définie sur  $\mathbb{R}$  par  $f(x) = 2x^3 - 3x + 1$  et on note  $\mathcal{C}_f$  sa courbe représentative dans un repère.1. Compléter l'algorithme suivant pour qu'il affiche si un point, dont on saisit les coordonnées, appartient ou non à la courbe  $\mathcal{C}_f$ .

Variables	$X$ et $Y$ sont des flottants
Entrée	Demandez la valeur de $X$
Traitement et sortie	<p>Si <math>Y = 2X^3 - 3X + 1</math>...</p> <p>Alors Afficher « Le point appartient à <math>\mathcal{C}_f</math> »</p> <p>Sinon Afficher « Le point n'appartient pas à <math>\mathcal{C}_f</math> »</p>
	Fin Si

2. On saisit -1 pour  $X$  et 2 pour  $Y$ . Qu'affiche l'algorithme ?L'algorithme affiche : « Le point appartient à  $\mathcal{C}_f$  ».3. Programmer cet algorithme. [www.](#)

**EXERCICE 57**

Fonctions : tableau de valeurs

Boucle bornée « POUR » (For) → p. 18

Comprendre un programme • Modifier un programme

On considère la fonction  $f$  définie sur l'intervalle  $[0 ; 10]$  par  $f(x) = x^2 + 5x + 3$ .

1. Quelles sont les valeurs affichées par le programme ci-dessous écrit en langage Python ?

3 ; 9 ; 17 ; 27 ; 39 ; 53 ; 69 ; 87 ; 107 ; 129 ; 153.

```
for i in range(0,11):
    Y=i**2+5*i+3
    print(Y)
```

2. Comment modifier l'algorithme précédent pour que celui-ci permette de compléter le tableau de valeurs ci-dessous ?

$x$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
$f(x)$											

On remplace l'instruction «  $Y=i**2+5*i+3$  » par l'instruction «  $Y=(0.1*i)**2+5*0.1*i+3$  ».

**EXERCICE 58**

Fonctions : généralités

Instruction conditionnelle (If) → p. 14

Comprendre un programme

Le programme ci-dessous, écrit en langage Python, permet de définir une fonction  $f$ .

1. Quelle valeur est renvoyée lorsqu'on saisit dans la console :

a.  $f(2)$ ?      b.  $f(0)$ ?      c.  $f(-5)$ ?      d.  $f(1,5)$ ?

a. La valeur renvoyée est 3.

b. La valeur renvoyée est 0.

c. La valeur renvoyée est 25.

d. La valeur renvoyée est 2.

2. a. Déterminer l'expression  $f(x)$  lorsque  $x \in ]-\infty ; 1[$ .

$$f(x) = x^2$$

- b. Déterminer l'expression  $f(x)$  lorsque  $x \in [1 ; +\infty[$ .

$$f(x) = 2x - 1$$

```
def f(X):
    if X<1:
        Y=X**2
    else:
        Y=2*X-1
    return(Y)
```

**EXERCICE 59**

Fonctions : généralités

Instruction conditionnelle (If) → p. 14

Écrire un algorithme • Programmer un algorithme

On considère la fonction  $f$  définie sur  $\mathbb{R}$  par :  $f(x) = -x + 1$  si  $x \leq 0$  et  $f(x) = x + 1$  si  $x > 0$ .

1. Calculer les images par  $f$  des réels  $-5 ; 2 ; -1,7 ; 3 ; 0$ .

$$f(-5) = 6 ; f(2) = 3 ; f(-1,7) = 2,7 ; f(3) = 4 \text{ et } f(0) = 1$$

2. Compléter l'algorithme ci-dessous pour qu'il affiche l'image par  $f$  d'un réel  $X$  saisi.

<b>Variables</b>	$X$ et $Y$ sont des flottants
<b>Entrée</b>	Demandez la valeur de $X$
<b>Traitement</b>	Si $X \leq 0$
	Alors $Y$ prend la valeur $-X + 1$
	Sinon $Y$ prend la valeur $X + 1$
	Fin Si
<b>Sortie</b>	Afficher $Y$

3. Programmer cet algorithme. [www.](#)

**EXERCICE 60**

Fonctions : maximum

Boucle non bornée « TANT QUE » (While) → p. 22

Compléter un algorithme • Programmer un algorithme

On considère la fonction  $f$  définie sur  $[0 ; 100]$  par  $f(x) = x^3 + x + 200$ .

On admet que la fonction  $f$  est croissante sur  $[0 ; 100]$ .

1. Quel est le maximum de  $f$  sur l'intervalle  $[0 ; 100]$  ?

Le maximum de  $f$  sur  $[0 ; 100]$  est  $f(100) = 1.000.300$ .

2. Compléter l'algorithme ci-dessous pour qu'il affiche le plus petit entier naturel  $n$  tel que  $f(n) \geq 15.000$ .

<b>Variables</b>	$N$ et $Y$ sont des entiers
<b>Initialisation</b>	$N$ prend la valeur 0
	$Y$ prend la valeur 200
<b>Traitement</b>	Tant que $Y \leq 15.000$ faire
	$N$ prend la valeur $N + 1$
	$Y$ prend la valeur $N^3 + N + 200$
	Fin Tant que
<b>Sortie</b>	Afficher $N$

3. Programmer cet algorithme. [www.](#)

4. Déterminer l'entier  $n$  cherché.

On obtient  $n = 25$ .

Instruction conditionnelle → p. 14  
Boucle bornée « POUR » (For) → p. 18

**EXERCICE 61**

Fonctions : maximum

Compléter un algorithme • Modifier et programmer un algorithme

On considère la fonction  $f$  définie sur l'intervalle  $[0 ; 20]$  par  $f(x) = x^3 - 27x^2 + 135x + 7$ .

On admet que le maximum de la fonction  $f$  est atteint pour un entier  $N$  de l'intervalle  $[0 ; 20]$ .

1. Compléter l'algorithme suivant pour qu'il affiche le maximum  $M$  de  $f$ .

<b>Variables</b>	$I$ est un entier
<b>Initialisation</b>	$M$ et $F$ sont des flottants
	$M$ prend la valeur 7
<b>Traitement</b>	Pour $I$ variant de 1 à 20 faire
	$F$ prend la valeur $I^3 - 27 \times I^2 + 135 \times I + 7$
	Si $F \geq M$
	Alors $M$ prend la valeur $F$
	Fin Si
	Fin Pour
<b>Sortie</b>	Afficher $M$

2. Comment modifier l'algorithme précédent pour qu'il affiche également l'entier  $N$  pour lequel  $f$  atteint son maximum ?

Il faut définir une autre variable entière  $N$ , l'initialiser à 0, ajouter dans l'instruction conditionnelle « Alors »... l'instruction «  $N$  prend la valeur  $I$  », puis ajouter en sortie l'instruction « Afficher  $N$  ».

3. a. Programmer ce deuxième algorithme. [www.](#)

- b. Déterminer le maximum  $M$  de la fonction  $f$  sur  $[0 ; 20]$  et l'entier  $N$  pour lequel il est atteint.

On obtient  $M = 196$  et  $N = 3$ .

## B Fonctions de référence

### EXERCICE 62

Fonction affine

Algorithme – Variables – Affectation → p. 3

Compléter un programme

1. Soit  $f$  la fonction définie sur  $\mathbb{R}$  par  $f(x) = -5x + 3$  et  $\mathcal{D}$  la droite représentative de  $f$ .a. Calculer  $f(0)$ . En déduire les coordonnées du point d'intersection de  $\mathcal{D}$  et de l'axe des ordonnées. $f(0) = -5 \times 0 + 3 = 3$ . Le point d'intersection de  $\mathcal{D}$  et de l'axe des ordonnées a pour coordonnées  $(0 ; 3)$ .b. Résoudre l'équation  $f(x) = 0$ . En déduire les coordonnées du point d'intersection de  $\mathcal{D}$  et de l'axe des abscisses. $-5x + 3 = 0$  équivaut à  $-5x = -3$ , soit  $x = 0,6$ .Le point d'intersection de  $\mathcal{D}$  et de l'axe des abscisses a pour coordonnées  $(0,6 ; 0)$ .2. Soit  $g$  la fonction définie sur  $\mathbb{R}$  par  $g(x) = ax + b$ , où  $a$  et  $b$  sont des réels non nuls.Soit  $(d)$  la droite représentative de  $g$ .Compléter le programme ci-dessous, écrit en langage Python, afin qu'il affiche les coordonnées des points d'intersection de  $(d)$  et des axes du repère, lorsque l'on saisit les valeurs de  $a$  et de  $b$ .

```
a=float(input("a="))
b=float(input("b="))
print(0..b..)
X=...b/a...
print(X,0)
```

### EXERCICE 63

Fonction affine

Instruction conditionnelle (If) → p. 14

Boucle non bornée « TANT QUE » (While) → p. 22

Fonctions → p. 26

Compléter un programme

Un magasin de location de scooters affiche les tarifs ci-dessous.

Moins de 8 heures : 1,60 € l'heure et 1,40 € de frais de dossier  
À partir de 8 heures : 1,40 € l'heure et 3 € de frais de dossier

1. Calculer le prix à payer pour :

a. 5 heures de location :  $5 \times 1,6 + 1,4 = 9,4$ . Prix pour 5 heures : 9,4 €.b. 20 heures de location :  $20 \times 1,4 + 3 = 31$ . Prix pour 20 heures : 31 €.2. On note  $P$  la fonction qui, au nombre  $x$  d'heures de location, associe le prix à payer (en euros).

Compléter les phrases suivantes :

– Si  $0 < x < 8$ , alors  $P(x) = 1,6x + 1,4$ .– Si  $x \geq 8$ , alors  $P(x) = 1,4x + 3$ .3. Dans le programme ci-dessous, écrit en langage Python, la fonction `prix` doit retourner le prix à payer pour  $x$  heures de location.

Compléter ce programme.

```
def prix(x):
    if x<8:
        p=1.6*x+1.4
    else
        p=1.4*x+3
    return(p)
```

```
h=0.5
while h<=24:
    print(...prix(h)...)
    h=h+0.5
```

4. En utilisant la fonction `prix` de la question précédente, compléter le programme ci-dessous afin qu'il affiche le prix à payer pour 30 minutes de location, pour 1 h, pour 1 h 30, pour 2 h, pour 2 h 30..., pour 23 h 30, pour 24 h.

### EXERCICE 64

Fonction affine

Comprendre un algorithme

Boucle bornée « POUR » (For) → p. 18

ABCD est un carré et M est un point qui se déplace sur le segment [AB].

On pose  $AM = x$ ,  $x$  appartenant à l'intervalle  $[0 ; 4]$ .On note  $V(x)$  l'aire de la surface colorée en vert et  $R(x)$  l'aire de la surface colorée en rouge.1. Donner l'expression de  $V(x)$  et celle de  $R(x)$  : $V(x) = 4x$  $R(x) = 4^2 - V(x) = 16 - 4x$ 

2. On donne un algorithme. Le faire fonctionner en complétant le tableau ci-dessous.

Variables	$V$ et $R$ sont des flottants $K$ est un entier
Traitement et sortie	<b>Pour</b> $K$ variant de 0 à 4 <b>faire</b> $V$ prend la valeur $4K$ $R$ prend la valeur $16 - V$ <b>Afficher</b> $V$ et $R$ <b>Fin Pour</b>
	<b>Fin de la 1<sup>re</sup> itération</b>
	<b>Fin de la 2<sup>e</sup> itération</b>
	<b>Fin de la 3<sup>e</sup> itération</b>
	<b>Fin de la 4<sup>e</sup> itération</b>
	<b>Fin de la 5<sup>e</sup> itération</b>

$K$	$V$	$R$
0	0	16
1	4	12
2	8	8
3	12	4
4	16	0

3. Quel est le rôle de cet algorithme ?

Cet algorithme permet d'afficher les valeurs de  $V(x)$  et de  $R(x)$  pour  $x$  variant de 0 à 4 avec un pas de 1.4. Par quelle instruction peut-on remplacer l'instruction : «  $R$  prend la valeur  $16 - V$  » ?On peut remplacer cette instruction par «  $R$  prend la valeur  $16 - 4K$  ».

### EXERCICE 65

Fonction affine

Boucle non bornée « TANT QUE » (While) → p. 22

Compléter un programme

Armelle souhaite travailler quelques heures par mois dans un musée, afin de gagner un peu d'argent. À la suite d'un entretien, deux possibilités d'indemnisation lui sont proposées.

Première proposition : 8 € par heure.

Deuxième proposition : versement de 91 € en début de mois, puis 5 € par heure.

Soit  $f$  et  $g$  les fonctions qui, au nombre  $x$  d'heures travaillées par mois, associent la somme que percevrait Armelle respectivement avec la première proposition et avec la deuxième proposition.

1. Si Armelle travaille 21 heures par mois, quelle est la proposition la plus intéressante pour elle ?

Avec la première proposition, l'indemnisation est égale à 168 € (car  $8 \times 21 = 168$ ).Avec la deuxième proposition, l'indemnisation est égale à 196 € (car  $91 + 5 \times 21 = 196$ ).

La deuxième proposition est donc la plus intéressante.

2. Donner, pour  $x$  appartenant à  $[1 ; 40]$ , l'expression de la fonction  $f$  et celle de la fonction  $g$ . $f(x) = 8x$  $g(x) = 91 + 5x$ 

3. Compléter le programme ci-dessous, écrit en langage Python, afin qu'il affiche le nombre d'heures à partir duquel la première proposition devient plus intéressante que la deuxième proposition.

```
P1=8
P2=96
k=1
while....P1<P2....:
    k=k+1
    P1=8*k
    P2=..91+5*k..
    print(..k..)
```

**EXERCICE 66**

Fonction polynôme de degré 2 | Algorithme – Variables – Affectation → p. 3

Comprendre un algorithme

On considère deux algorithmes.

**Algorithme 1**

<b>Variables</b>	$X$ et $Y$ sont des flottants
<b>Entrée</b>	Demander la valeur de $X$
<b>Traitements</b>	Affecter à $Y$ la valeur $X + 2$ Affecter à $Y$ la valeur $Y^2 - 1$
<b>Sortie</b>	Afficher $Y$

1. Lorsque l'on saisit  $X = 3$ , quelle valeur est affichée en sortie :

a. avec l'algorithme 1 ?

$$(3+2)^2-1=24$$

b. avec l'algorithme 2 ?

$$3^2+4\times 3+3=24$$

2. Quel que soit le nombre saisi, la valeur affichée en sortie par l'algorithme 1 est-elle la même que celle affichée par l'algorithme 2 ? Justifier.

Oui, car si le nombre saisi est  $a$ , on a avec l'algorithme 1 :  $(a+2)^2-1=a^2+4a+3$   
..... avec l'algorithme 2 :  $a^2+4a+3$ **EXERCICE 67**

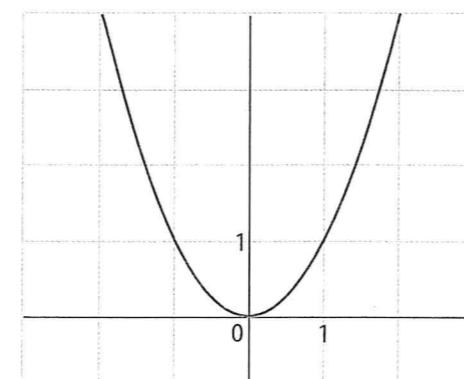
Fonction carré

Instruction conditionnelle (If) → p. 14

Compléter un programme

1. La courbe ci-contre est celle de la fonction carré sur l'intervalle  $[-2 ; 2]$ .Dans chaque cas, déterminer le minimum  $m$  et le maximum  $M$  de la fonction carré sur l'intervalle donné.a. Sur  $[1 ; 2]$  :  $m = \dots$  et  $M = \dots$ b. Sur  $[-2 ; 1]$  :  $m = \dots$  et  $M = \dots$ c. Sur  $[-2 ; -1]$  :  $m = \dots$  et  $M = \dots$ 2. On note  $m$  le minimum et  $M$  le maximum de la fonction carré sur un intervalle  $[a ; b]$  donné (avec  $a < b$ ).

Dire si les phrases suivantes sont vraies ou fausses.

a.  $M$  est le plus grand des deux nombres  $a^2$  et  $b^2$ . Vrai Fauxb.  $m$  est le plus petit des deux nombres  $a^2$  et  $b^2$ . Vrai Fauxc. Si  $a$  et  $b$  sont de signes contraires,  $m$  est égal à 0. Vrai Faux3. Compléter le programme ci-contre afin qu'il affiche la valeur de  $m$  et celle de  $M$  lorsque l'on saisit  $a$  et  $b$  (avec  $a < b$ ).Dans ce programme, écrit en langage Python, on utilise les fonctions `max` et `min`, intégrées dans le logiciel, qui retournent respectivement le plus grand et le plus petit de deux nombres donnés en arguments.

```

a=float(input("a="))
b=float(input("b="))
M=max(.....,a*a,b*b.....)
if a<0 and b>0:
    m=..0..
else
    m=..min(a*a,b*b)..
print("Le minimum est : ",m)
print("Le maximum est : ",M)

```

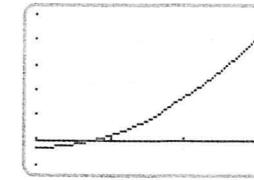
© Bordas 2017 - La photocopie non autorisée est un délit.

**EXERCICE 68**

Fonction polynôme de degré 2

Instruction conditionnelle (If) → p. 14  
Boucle non bornée « TANT QUE » (While) → p. 22

Comprendre et compléter un algorithme • Programmer un algorithme

On considère la fonction  $f$  définie sur l'intervalle  $[5 ; 20]$  par  $f(x) = x^2 - 10x + 10$ .On a tracé ci-contre, à l'aide d'une calculatrice, la courbe représentative de  $f$ .  
1. Justifier, à l'aide du graphique, que l'équation  $f(x) = 0$  a une unique solution dans l'intervalle  $[5 ; 20]$ . On note  $x_0$  cette solution.La courbe coupe une seule fois l'axe des abscisses, donc l'équation a une unique solution dans  $[5 ; 20]$ .2. a. Déterminer le signe de  $f(5) \times f(6)$ .

$$f(5) = -15 \text{ et } f(6) = -14 \text{ donc } f(5) \times f(6) \geq 0$$

b.  $x_0$  appartient-il à l'intervalle  $[5 ; 6]$  ?Non,  $x_0$  n'appartient pas à l'intervalle  $[5 ; 6]$ .

3. On considère l'algorithme ci-contre.

a. Faire fonctionner cet algorithme en complétant le tableau ci-dessous avec  $a = 5$  et  $p = 1$  comme entrées.

Variables	$a$ et $b$ sont des flottants
Entrée	Demander les valeurs de $a$ et de $p$
Traitements et sortie	$b$ prend la valeur $a + p$ Tant que $f(a) \times f(b) > 0$ faire   $a$ prend la valeur $b$   $b$ prend la valeur $a + p$ Fin Tant que Afficher $a$ et $b$

	$a$	$b$	Condition $f(a) \times f(b) > 0$
Avant le début de la boucle	5	6	Vrai car : $f(5) = -15$ et $f(6) = -14$ donc $f(5) \times f(6) > 0$ .
Après le 1 <sup>er</sup> passage dans la boucle	6	7	Vrai car : $f(6) = -14$ et $f(7) = -11$ donc $f(6) \times f(7) > 0$ .
Après le 2 <sup>e</sup> passage dans la boucle...	7	8	Vrai car : $f(7) = -11$ et $f(8) = -6$ donc $f(7) \times f(8) > 0$ .
Après le 3 <sup>e</sup> passage dans la boucle...	8	9	Faux car : $f(8) = -6$ et $f(9) = 1$ donc $f(8) \times f(9) \leq 0$ .

b. Quelles sont les valeurs affichées en sortie ?

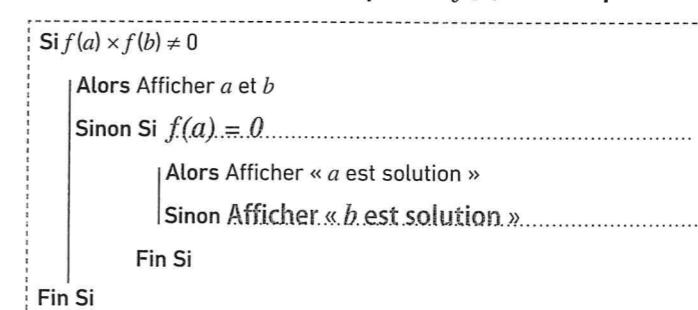
Les valeurs affichées en sortie sont 8 et 9.

c. Quel encadrement de  $x_0$  peut-on en déduire ?On en déduit que  $8 \leq x_0 \leq 9$ .4. a. Programmer cet algorithme. [www.](#)b. Utiliser ce programme pour donner un encadrement de  $x_0$  d'amplitude  $p = 0,1$ .On saisit  $a = 8$  et  $p = 0,1$ . On obtient  $8,8 \leq x_0 \leq 8,9$ .5. Dans cette question,  $f$  est la fonction définie sur l'intervalle  $[5 ; 20]$  par  $f(x) = x^2 - 10x + 21$ .

a. Quelles sont les valeurs affichées en sortie par l'algorithme ?

Les valeurs affichées en sortie sont 6 et 7, car  $f(6) = -3$  et  $f(7) = 0$  donc  $f(6) \times f(7) \leq 0$ .b. Vérifier que 7 est une solution de l'équation  $f(x) = 0$ .

$$7^2 - 10 \times 7 + 21 = 0, \text{ donc } 7 \text{ est bien solution de l'équation } f(x) = 0.$$

c. Compléter la suite d'instructions ci-dessous, destinée à remplacer la dernière instruction « Afficher  $a$  et  $b$  » de l'algorithme, afin que soit affichée la solution de l'équation  $f(x) = 0$  lorsque c'est possible.

**EXERCICE 69**

Fonction polynôme de degré 2

Instruction conditionnelle (if) → p. 14

Comprendre un programme

On considère la fonction  $g$  définie sur  $\mathbb{R}$  par  $g(x) = 2x^2 + 3x + 1$ .1. Pour quelle valeur de  $x$  la fonction  $g$  admet-elle un extremum ? $g(x)$  est de la forme  $ax^2 + bx + c$  avec  $a = 2$ ,  $b = 3$  et  $c = 1$ . $\frac{-b}{2a} = \frac{-3}{2 \times 2} = -0,75$ . La fonction  $g$  admet un extremum en  $-0,75$ .

2. Cet extremum est-il un maximum ou un minimum ?

C'est un minimum car le coefficient  $a$  est positif ( $a = 2$ ). La fonction  $g$  est donc décroissante, puis croissante.

3. Quel est le rôle du programme ci-dessous, écrit en langage Python ?

```

a=2
b=3
c=1
x=-b/(2*a)
m=a*x*x+b*x+c
k=float(input("k ="))
if k>m :
    print("2 solutions")
if k==m :
    print("une seule solution")
if k<m :
    print("pas de solution")

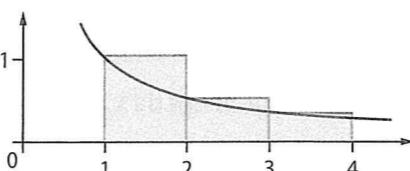
```

Ce programme affiche, lorsque l'on entre la valeur d'un nombre réel  $k$ , le nombre de solutions de l'équation $g(x) = k$ .**EXERCICE 70**

Fonction inverse

Boucle bornée « POUR » (For) → p. 18

Compléter et programmer un algorithme

Dans un repère orthonormé d'unité graphique 1 cm, on a tracé la courbe représentative de la fonction inverse sur l'intervalle  $[0; +\infty[$ . On construit  $n$  rectangles de largeur 1 cm et de hauteurs  $1 \text{ cm}, \frac{1}{2} \text{ cm}, \frac{1}{3} \text{ cm}, \dots, \frac{1}{n} \text{ cm}$ .La figure ci-contre a été réalisée pour  $n = 3$ .1. Dans chacun des cas suivants, calculer la valeur exacte de l'aire (en  $\text{cm}^2$ ) de la surface formée par  $n$  rectangles :a. pour  $n = 3$  :  $1 \times 1 + 1 \times \frac{1}{2} + 1 \times \frac{1}{3} = 1 + \frac{1}{2} + \frac{1}{3} = \frac{11}{6}$ . L'aire est égale à  $\frac{11}{6} \text{ cm}^2$ .b. pour  $n = 4$  :  $\frac{11}{6} + \frac{1}{4} = \frac{25}{12}$ . L'aire est égale à  $\frac{25}{12} \text{ cm}^2$ .2. On donne un algorithme. Le compléter par l'une des instructions ci-dessous afin qu'il affiche l'aire de la surface formée par  $n$  rectangles. Programmer cet algorithme. [www.](#) $S$  prend la valeur  $\frac{1}{K}$  $S$  prend la valeur  $S + \frac{1}{n}$  $S$  prend la valeur  $S + \frac{1}{K}$  $S$  prend la valeur  $S + \frac{1}{S}$ 

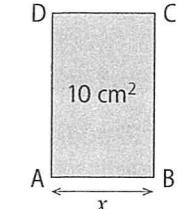
<b>Variables</b>	$S$ est un flottant. $K$ et $n$ sont des entiers
<b>Entrée</b>	Saisir $n$
<b>Initialisation</b>	$S$ prend la valeur 0
<b>Traitement</b>	Pour $K$ variant de 1 à $n$ faire  ..... $S$ prend la valeur $S + \frac{1}{K}$ .....
<b>Sortie</b>	Afficher $S$

**EXERCICE 71**

Fonction inverse

Boucle non bornée « TANT QUE » (While) → p. 22

Comprendre un algorithme • Écrire un programme

ABCD est un rectangle dont l'aire est égale à  $10 \text{ cm}^2$ .On note  $AB = x$ , avec  $x$  appartenant à l'intervalle  $[1 ; 10]$ .Soit  $f$  la fonction qui à tout réel  $x$  de  $[1 ; 10]$  associe la longueur  $BC$ .1. Donner l'expression de  $f$ . $f(x) = \frac{10}{x}$ 

2. a. On donne un algorithme. Le faire fonctionner en complétant le tableau ci-dessous.

Variables	$X$ et $Y$ sont des flottants
<b>Initialisation</b>	$X$ prend la valeur 1 $Y$ prend la valeur $\frac{10}{X}$
<b>Traitement</b>	Tant que $Y \geq 2$ faire  ..... $X$ prend la valeur $X + 1$ .....  ..... $Y$ prend la valeur $\frac{10}{X}$ .....
<b>Sortie</b>	Fin Tant que Afficher $X$

	$X$	$Y$	Condition $Y \geq 2$
Avant le début de la boucle	1	10	Vrai car $10 \geq 2$
Après le 1 <sup>er</sup> passage dans la boucle	2	5	Vrai car $5 \geq 2$
Après le 2 <sup>e</sup> passage dans la boucle	3	$\approx 3,3$	Vrai car $3,3 \geq 2$
Après le 3 <sup>e</sup> passage dans la boucle	4	$\approx 2,5$	Vrai car $2,5 \geq 2$
Après le 4 <sup>e</sup> passage dans la boucle	5	$\approx 2$	Vrai car $2 \geq 2$
Après le 5 <sup>e</sup> passage dans la boucle	6	$\approx 1,7$	Faux car $1,7 < 2$

b. Quelle est la valeur affichée en sortie ?

La valeur affichée est 6.

c. Quel est le rôle de cet algorithme ?

Cet algorithme affiche la valeur entière de  $x$  à partir de laquelle la longueur  $BC$  devient strictement inférieure à 2 cm.3. Écrire un programme qui affiche, au millimètre près, la valeur de  $x$  à partir de laquelle la longueur  $BC$  est strictement inférieure à la longueur  $AB$ . Quelle est la valeur affichée ? [www.](#)

La valeur affichée est : environ 3,2.

**EXERCICE 72**

Fonction inverse

Boucle bornée « POUR » (For) → p. 18

Comprendre un programme

Soit  $f$  et  $g$  les fonctions définies sur l'intervalle  $[1 ; +\infty[$  par  $f(x) = \frac{1}{x}$  et  $g(x) = 5 - x$ .1. Compléter les tableaux de valeurs des fonctions  $f$  et  $g$  ci-dessous (arrondir les valeurs à 0,01 près).

$x$	1	2	3	4	5	6	7	8	9
$f(x)$	1	0,5	0,33	0,25	0,2	0,17	0,14	0,13	0,11
$g(x)$	4	3	2	1	0	-1	-2	-3	-4

2. On donne ci-contre un programme écrit en langage Python.

a. Qu'affiche ce programme lorsque l'on saisit  $b = 9$  ?

Ce programme affiche 1, 2, 3 et 4.

b. Quel est le rôle de ce programme ?

Ce programme affiche les solutions entières de l'inéquation  $\frac{1}{x} < 5 - x$  sur l'intervalle  $[1 ; b]$ ,  $b$  étant un entier naturel saisi par l'utilisateur.

```

b=int(input("b ="))
for a in range(1,b+1):
    y=1/a
    z=5-a
    if y<z:
        print(a)
    
```

## A Géométrie plane

### EXERCICE 73

Repérage du plan

Algorithmme – Variables – Affectation → p. 3

Comprendre un algorithme • Programmer un algorithme

1. Rappeler les formules permettant de calculer les coordonnées du milieu d'un segment [AB] dans un repère à partir de celles des points A et B.

Si I est le milieu du segment [AB],  $x_I = \frac{x_A + x_B}{2}$  et  $y_I = \frac{y_A + y_B}{2}$ .

2. On considère l'algorithme suivant :

Variables	$x_B, y_B, x$ et $y$ sont des flottants
Entrée	Saisir $x_B$ et $y_B$
Traitement	$x$ prend la valeur $\frac{5 + x_B}{2}$ $y$ prend la valeur $\frac{-7 + y_B}{2}$
Sortie	Afficher $x$ , $y$

- a. On entre 3 pour  $x_B$  et 1 pour  $y_B$ . Quelles sont les valeurs affichées en sortie ?

Les valeurs affichées en sortie sont 4 pour  $x$  et -3 pour  $y$ .

- b. Interpréter géométriquement les valeurs affichées en sortie dans le cas général.

$x$  est l'abscisse du milieu du segment [AB] où A est le point de coordonnées (5 ; -7) et B le point de coordonnées ( $x_B$  ;  $y_B$ ),  $y$  est l'ordonnée de ce milieu.

3. Programmer cet algorithme. [www.](#)

### EXERCICE 74

Repérage du plan

Instruction conditionnelle (If) → p. 14  
Fonctions → p. 26

Comprendre un programme • Écrire un programme

Soit A le point de coordonnées (-1 ; 2) dans un repère orthonormé.

1. On considère le programme de la fonction  $L$  ci-contre écrit en langage Python.

```
def L(xB,yB):
    d=sqrt((xB+1)**2+(yB-2)**2)
    return(d)
```

- a. On saisit  $L(7,17)$  dans la console. Quelle est la valeur rentrée ?

La valeur rentrée est 17.

- b. Quel est le rôle de cette fonction ?

Cette fonction permet d'obtenir la distance AB où B est le point de coordonnées ( $x_B$  ;  $y_B$ ) dans le repère orthonormé.

2. On considère le cercle  $\mathcal{C}$  de centre A et de rayon 5.

- a. Le point E de coordonnées (7 ; 17) appartient-il au cercle  $\mathcal{C}$  ? Justifier.

Le point n'appartient pas au cercle  $\mathcal{C}$  car  $AE = 17$ , donc  $AE \neq 5$ .

- b. En utilisant la fonction précédente, écrire un programme qui, à partir de la saisie des coordonnées  $x_B$  et  $y_B$  du point B, affiche « oui » si B appartient au cercle  $\mathcal{C}$  et « non » si B n'appartient pas à ce cercle. [www.](#)

```
xB=float(input("xB="))
yB=float(input("yB="))
if...L(xB,yB)==5:
    print("oui")
else:
    print("non")
```

### EXERCICE 75

Configurations du plan

Boucle bornée « POUR » (For) → p. 18

Compléter un algorithme • Programmer un algorithme

Dans un repère du plan, on considère les points A(2 ; 5) et M<sub>0</sub>(6 ; 9).

1. Déterminer les coordonnées du point M<sub>1</sub> milieu du segment [AM<sub>0</sub>].

M<sub>1</sub> a pour coordonnées (4 ; 7).

2. On note M<sub>2</sub> le milieu du segment [AM<sub>1</sub>].

Déterminer les coordonnées du point M<sub>2</sub>.

M<sub>2</sub> a pour coordonnées (3 ; 6).

3. On poursuit le processus : M<sub>3</sub> est le milieu de [AM<sub>2</sub>] puis M<sub>4</sub> est le milieu de [AM<sub>3</sub>], etc.

Compléter l'algorithme ci-contre pour qu'il affiche les coordonnées du point M<sub>12</sub>.

4. Programmer cet algorithme. [www.](#)

Variables	$I$ est un entier
Initialisation	$x_M$ et $y_M$ sont des flottants $x_M$ prend la valeur 6 $y_M$ prend la valeur 9
Traitement	Pour $I$ variant de 1 à 12 faire $x_M$ prend la valeur $\frac{2 + x_M}{2}$ $y_M$ prend la valeur $\frac{5 + y_M}{2}$
Fin Pour	
Sortie	Afficher $x_M$ , $y_M$

### EXERCICE 76

Configurations du plan

Boucle non bornée « TANT QUE » (While) → p. 22

Analyser une situation • Compléter un programme

On considère un triangle OM<sub>0</sub>M<sub>1</sub> isocèle rectangle en M<sub>0</sub> tel que OM<sub>0</sub> = 10 cm. On note T<sub>1</sub> ce triangle.

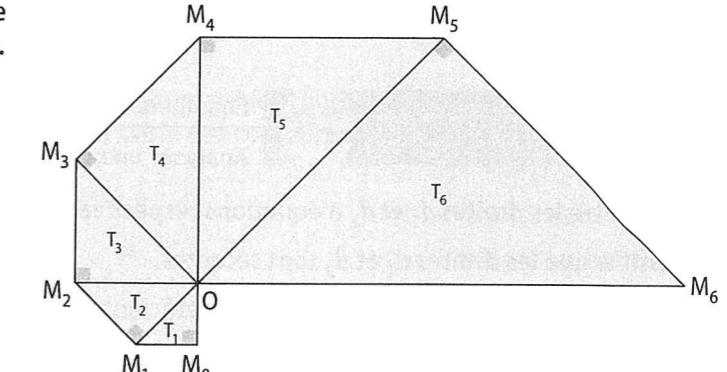
On construit ensuite le triangle OM<sub>1</sub>M<sub>2</sub> rectangle isocèle en M<sub>1</sub>. On note T<sub>2</sub> ce deuxième triangle.

On continue la construction et on obtient les triangles T<sub>3</sub>, T<sub>4</sub>... (voir figure ci-contre).

1. a. Justifier l'égalité OM<sub>1</sub> =  $\sqrt{200}$ .

OM<sub>1</sub> est la longueur de l'hypoténuse du triangle rectangle isocèle OM<sub>0</sub>M<sub>1</sub>.

On a donc OM<sub>1</sub><sup>2</sup> = OM<sub>0</sub><sup>2</sup> + M<sub>1</sub>M<sub>0</sub><sup>2</sup> soit OM<sub>1</sub><sup>2</sup> = 10<sup>2</sup> + 10<sup>2</sup> = 200 et OM<sub>1</sub> =  $\sqrt{200}$ .



- b. En déduire l'aire  $\mathcal{A}_2$  en cm<sup>2</sup> du triangle T<sub>2</sub>. Quel est le rapport  $\frac{\mathcal{A}_2}{\mathcal{A}_1}$  où  $\mathcal{A}_1$  est l'aire en cm<sup>2</sup> du triangle T<sub>1</sub> ?

$$\mathcal{A}_2 = \frac{\sqrt{200} \times \sqrt{200}}{2} = \frac{200}{2} = 100 \text{ et } \mathcal{A}_1 = \frac{10 \times 10}{2} = 50. \text{ On a } \frac{\mathcal{A}_2}{\mathcal{A}_1} = 2.$$

2. On admet que l'aire de chaque triangle est le double de l'aire du triangle précédent.

- a. Convertir 5 m<sup>2</sup> en cm<sup>2</sup>.

5 m<sup>2</sup> est équivalent à 50 000 cm<sup>2</sup>.

- b. Compléter le programme ci-contre, écrit en langage Python, pour qu'il affiche le numéro N du premier triangle dont l'aire est supérieure ou égale à 5 m<sup>2</sup>.

```
A=50
N=1
While....A<50.000....:
    A=..2*A
    N=..N+1
    print(..N..)
```

**EXERCICE 77**

Équations de droites

Compléter un algorithme • Programmer un algorithme

1. Compléter l'algorithme suivant pour qu'il affiche lorsqu'il existe le coefficient directeur de la droite (AB) à partir de la saisie des coordonnées des points A et B.

<b>Variables</b>	$x_A, y_A, x_B, y_B$ et $a$ sont des flottants
<b>Entrée</b>	Saisir $x_A, y_A, x_B$ et $y_B$
<b>Traitement et sortie</b>	<p>Si ..... <math>x_A = x_B</math>.....</p> <p>Alors Afficher « La droite (AB) n'a pas de coefficient directeur »</p> <p>Sinon Affecter à <math>a</math> la valeur ..... <math>\frac{y_B - y_A}{x_B - x_A}</math>.....</p> <p>Afficher <math>a</math></p>
	Fin Si

2. Programmer cet algorithme. [www.](#)

**EXERCICE 78**

Équations de droites

Instruction conditionnelle (If) → p. 14

Écrire un algorithme

On considère la droite  $d_1$  d'équation  $y = 2x + 3$  et la droite  $d_2$  d'équation  $y = ax + 5$  où  $a$  est un nombre réel. Écrire un algorithme permettant d'afficher si les droites  $d_1$  et  $d_2$  sont sécantes ou non à partir de la saisie du réel  $a$ .

<b>Variable</b>	$a$ est un flottant
<b>Entrée</b>	Saisir $a$
<b>Traitement et sortie</b>	<p>Si ..... <math>a \neq 2</math>.....</p> <p>Alors ..... Afficher « <math>d_1</math> et <math>d_2</math> sont sécantes ».....</p> <p>Sinon ..... Afficher « <math>d_1</math> et <math>d_2</math> ne sont pas sécantes ».....</p>
	Fin Si

**EXERCICE 79**

Équations de droites

Algorithmme – Variables – Affectation → p. 3  
Fonctions → p. 26

Analyser une situation • Écrire un programme

On considère les droites  $d_1$  et  $d_2$  d'équations respectives  $y = 3x + b_1$  et  $y = 2x + b_2$ .

1. Justifier que les droites  $d_1$  et  $d_2$  sont sécantes.

Les droites  $d_1$  et  $d_2$  sont sécantes car leurs coefficients directeurs respectifs ne sont pas égaux.

2. Soit  $(x_I ; y_I)$  le couple des coordonnées du point I, intersection des droites  $d_1$  et  $d_2$ . Montrer que  $x_I = b_2 - b_1$ .

On a  $y_I = 3x_I + b_1$  et  $y_I = 2x_I + b_2$ , donc  $3x_I + b_1 = 2x_I + b_2$  et  $x_I = b_2 - b_1$ .

3. Programmer une fonction d'arguments  $b_1$  et  $b_2$  qui retourne les coordonnées du point I. [www.](#)

```
def...coordonnées(b1,b2):.....
....xI=b2-b1.....
....yI=3*xI+b1.....
....return(xI,yI).....
```

**EXERCICE 80**

Équations de droites

Boucle bornée « POUR » (For) → p. 18

Compléter un algorithme

Soit A(2 ; 4) et B(-6 ; 0) deux points du plan.

1. Déterminer une équation de la droite (AB).

A et B n'ayant pas la même abscisse, la droite (AB) a pour équation  $y = ax + b$  avec  $a = \frac{0 - 4}{-6 - 2} = \frac{1}{2}$ .

Comme A ∈ (AB),  $4 = \frac{1}{2} \times 2 + b$  soit  $4 = 1 + b$  et  $b = 3$ . La droite (AB) a donc pour équation  $y = \frac{1}{2}x + 3$ .

2. Déterminer l'ordonnée du point C de la droite (AB) ayant pour abscisse -2.

L'ordonnée du point C est 2.

3. Compléter l'algorithme ci-dessous pour qu'il affiche l'ordonnée de chaque point de la droite (AB) d'abscisse entière comprise entre 0 et 8.

<b>Variables</b>	$I$ est un entier
<b>Entrée</b>	$y$ est un flottant
<b>Traitement</b>	Pour $I$ variant de ..... 0 ..... à ..... 8 ..... faire
	$y$ prend la valeur ..... $0,5 \times I + 3$ .....
	Afficher $y$
	Fin Pour

**EXERCICE 81**

Vecteurs

Écrire un programme

Algorithmme – Variables – Affectation → p. 3  
Fonctions → p. 26

Soit A(1 ; 3), B(4 ; 7), C( $x_C$  ;  $y_C$ ) et D( $x_D$  ;  $y_D$ ) quatre points du plan.

1. On suppose dans cette question que  $x_C = -2$  et  $y_C = 5$ .

Déterminer les coordonnées du point D pour que le quadrilatère ABDC soit un parallélogramme.

$\vec{AB}(3 ; 4)$  et  $\vec{CD}(x_D + 2 ; y_D - 5)$ . ABDC est un parallélogramme si et seulement si  $\vec{AB} = \vec{CD}$ .

c'est-à-dire si et seulement si  $\begin{cases} x_D + 2 = 3 \\ y_D - 5 = 4 \end{cases}$  d'où  $x_D = 1$  et  $y_D = 9$ .

2. Démontrer que le quadrilatère ABDC est un parallélogramme si et seulement si  $x_D = x_C + 3$  et  $y_D = y_C + 4$ .

$\vec{CD}(x_D - x_C ; y_D - y_C)$ . donc ABDC est un parallélogramme si et seulement si  $\begin{cases} x_D - x_C = 3 \\ y_D - y_C = 4 \end{cases}$  ce qui équivaut à  $\begin{cases} x_D = x_C + 3 \\ y_D = y_C + 4 \end{cases}$ .

3. Écrire le programme d'une fonction d'arguments  $x_C$  et  $y_C$  retournant les coordonnées du point D tel que le quadrilatère ABDC soit un parallélogramme à partir de la saisie des coordonnées du point C. [www.](#)

```
def...parallélogramme(xC,yC):.....
....xD=xC+3.....
....yD=yC+4.....
....return(xD,yD).....
```

**EXERCICE 82**

Vecteurs

Algorithmme – Variables – Affectation → p. 3

Écrire un algorithme

Dans un repère, on considère les vecteurs  $\vec{u}(-3 ; 5)$  et  $\vec{v}(a ; b)$  où  $a$  et  $b$  sont deux réels.

Pour tout réel  $k$  on définit le vecteur  $\vec{w} = \vec{u} + k\vec{v}$ .

1. Déterminer en fonction de  $a$ ,  $b$  et  $k$  les coordonnées du vecteur  $\vec{w}$ .

$\vec{w}(-3 + ka ; 5 + kb)$ .

2. En utilisant la structure ci-dessous, écrire un algorithme permettant d'afficher les coordonnées du vecteur  $\vec{w}$  à partir de la saisie des coordonnées du vecteur  $\vec{v}$  et du réel  $k$ .

<b>Variables</b>	$x_w, y_w, a, b$ et $k$ sont des flottants
<b>Entrée</b>	Saisir $a$ , $b$ et $k$
<b>Traitement</b>	$x_w$ prend la valeur $-3 + k \times a$ .....
<b>Sortie</b>	$y_w$ prend la valeur $5 + k \times b$ .....

**B Géométrie dans l'espace****EXERCICE 83**

Géométrie dans l'espace

Algorithme – Variables – Affectation → p. 3

Comprendre un algorithme • Modifier un algorithme

Une boîte en forme de pavé droit a pour dimensions  $a$ ,  $b$  et  $c$ .  
On donne l'algorithme ci-dessous.

1. On saisit  $a = 3$ ,  $b = 5$  et  $c = 10$ . Quel est l'affichage en sortie ?  
 $3 \times 5 \times 10 = 150$ .

2. Que fait cet algorithme ? Il calcule le volume de la boîte à partir des dimensions de ses arêtes.

3. Modifier cet algorithme afin qu'il affiche en sortie l'aire de la surface latérale de cette boîte.

On remplace l'instruction du traitement par : « Affecter à  $X$  la valeur  $2 \times a \times b + 2 \times a \times c + 2 \times b \times c$  ».

<b>Variables</b>	$a, b, c, X$ sont des flottants
<b>Entrée</b>	Saisir $a, b, c$
<b>Traitement</b>	Affecter à $X$ la valeur $a \times b \times c$
<b>Sortie</b>	Afficher $X$

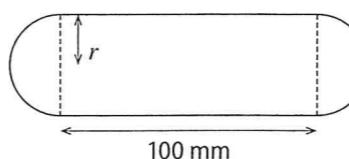
**EXERCICE 84**

Géométrie dans l'espace

Boucle non bornée « TANT QUE » (While) → p. 22

Analyser une situation • Compléter un algorithme

Une ampoule destinée à recevoir du sérum est constituée d'un corps cylindrique de hauteur 100 mm et de deux demi-sphères de rayon  $r$  millimètres. On veut déterminer à partir de quelle valeur entière du rayon, exprimé en millimètres, le volume de l'ampoule dépasse 20 centilitres.



1. Montrer que le volume de l'ampoule, exprimé en  $\text{mm}^3$ , est égal à  $100\pi r^2 + \frac{4}{3}\pi r^3$ .

Le volume du cylindre en  $\text{mm}^3$  est :

$$100 \times \pi r^2$$

Le volume de chaque demi-sphère

en  $\text{mm}^3$  est  $\frac{1}{2} \times \frac{4}{3}\pi r^3$ , d'où le résultat

annoncé.

2. Compléter l'algorithme ci-dessous afin qu'il réponde au problème posé.

<b>Variables</b>	$R$ est un entier $V$ est un flottant
<b>Initialisation</b>	$R$ prend la valeur 0 $V$ prend la valeur 0
<b>Traitement</b>	Tant que $V \leq 200\ 000$ faire   $R$ prend la valeur $R + 1$   $V$ prend la valeur $100 \times \pi \times R^2 + \pi \times R^3 \times \frac{4}{3}$
<b>Sortie</b>	Fin Tant que Afficher $R$

3. Programmer cet algorithme et répondre à la question posée. [www.](#)

Le programme affiche 23. Donc le volume de l'ampoule dépasse 20 centilitres pour un rayon de 23 mm.

**EXERCICE 85**

Géométrie dans l'espace

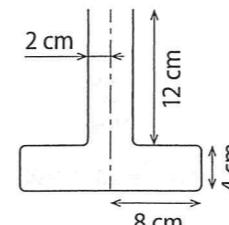
Instruction conditionnelle (If) → p. 14

Analyser une situation • Compléter un programme

Une carafe est constituée d'un cylindre de révolution de hauteur 4 cm et de rayon 8 cm, surmonté d'un autre cylindre de révolution de hauteur 12 cm et de rayon 2 cm.

1. Calculer, en  $\text{cm}^3$ , le volume d'eau dans la carafe quand la hauteur d'eau est  $x$  centimètres, où  $x$  est un nombre réel compris entre 0 et 4.

Le volume d'eau est, en  $\text{cm}^3$  :  $\pi \times 8^2 \times x = 64\pi x$ .



2. Montrer que le volume d'eau dans la carafe quand la hauteur d'eau dans la carafe est  $x$  centimètres, où  $x$  est un nombre réel compris entre 4 et 16, est égal à  $4\pi x + 240\pi$  (en  $\text{cm}^3$ ).

La hauteur d'eau dans le cylindre supérieur est  $x - 4$ , donc le volume

d'eau est, en  $\text{cm}^3$  :  $256\pi + \pi \times 2^2 \times (x - 4) = 4\pi x + 240\pi$ .

3. Compléter le programme de la fonction `vol_eau` ci-dessous, écrit en langage Python, afin qu'elle retourne le volume d'eau dans la carafe (en  $\text{cm}^3$ ) selon la hauteur d'eau  $H$  versée.

```
def vol_eau(..., H,...):
    if ..., H <= 4...:
        return(..., 64 * pi * H,...)
    else:
        return(..., 4 * pi * H + 240 * pi,...)
```

**A Statistiques****EXERCICE 86**

Statistiques

Algorithme – Variables – Affectation → p. 3

Fonctions → p. 26

Compléter un programme

Dans une résidence universitaire, il y a 150 chambres de  $12 \text{ m}^2$  et 250 chambres de  $18 \text{ m}^2$ .

1. Calculer la superficie moyenne (en  $\text{m}^2$ ) d'une chambre de cette résidence.

$$\frac{150 \times 12 + 250 \times 18}{400} = 15,75 \text{ m}^2 \text{ la superficie moyenne est de } 15,75 \text{ m}^2$$

2. a. Compléter le programme ci-dessous, écrit en langage Python, afin que la fonction `moyenne_2` retourne la moyenne de deux valeurs  $X1$  et  $X2$  d'effectifs respectifs  $N1$  et  $N2$ .

b. Comment peut-on afficher avec cette fonction `moyenne_2` la superficie moyenne calculée dans la question 1 ?

On peut, soit demander `moyenne_2(12, 150, 18, 250)` dans la console, soit utiliser l'instruction `print(moyenne_2(12, 150, 18, 250))`.

```
def moyenne_2(X1, N1, X2, N2):
    N = ... N1+N2 ...
    S = ... N1*X1+N2*X2 ...
    M = S/N
    return(M)
```

**EXERCICE 87**

Statistiques

Boucle non bornée « TANT QUE » (While) → p. 22

Comprendre un algorithme • Programmer un algorithme

On considère la série statistique : 5 - 7 - 7 - 8 - 9 - 9 - 9 - 12 - 14 - 14, dont les valeurs ont été ordonnées.

1. a. On donne un algorithme dans lequel l'utilisateur saisit un nombre  $Q$  puis, dans l'ordre croissant et tant que c'est nécessaire, les valeurs  $V$  de la série.

Le nombre  $Q$  doit être une valeur de la série différente de 14.

Faire fonctionner cet algorithme en complétant le tableau ci-dessous avec en entrées  $Q = 8$  et  $V = 5$ .

<b>Variables</b>	$Q$ et $V$ sont des flottants $N$ est un entier
<b>Initialisation</b>	$N$ prend la valeur 0
<b>Entrée</b>	Saisir $Q$ et $V$
<b>Traitement</b>	Tant que $V \leq Q$ faire   $N$ prend la valeur $N + 1$   Saisir $V$
<b>Sortie</b>	Fin Tant que Afficher $N$

	$N$	$V$	Condition $V \leq Q$
Avant le début de la boucle	0	5	Vrai car $5 \leq 8$
Après le 1 <sup>er</sup> passage dans la boucle	1	7	Vrai car $7 \leq 8$
Après le 2 <sup>e</sup> passage dans la boucle	2	7	Vrai car $7 \leq 8$
Après le 3 <sup>e</sup> passage dans la boucle	3	8	Vrai car $8 \leq 8$
Après le 4 <sup>e</sup> passage dans la boucle	4	9	Faux car $9 > 8$

b. Quelle est la valeur affichée en sortie ?

La valeur affichée est 4.

c. Comment peut-on interpréter cette valeur par rapport à la série statistique de l'énoncé ?

Il y a 4 valeurs de la série inférieures ou égales à 8.

2. a. Programmer cet algorithme. [www.](#)

b. Utiliser ce programme pour compléter le tableau des effectifs cumulés croissants ci-dessous :

<b>Valeurs</b>	5	7	8	9	12	14
<b>Effectifs cumulés croissants</b>	1	3	4	7	8	10

## B Échantillonnage et probabilités

### EXERCICE 88

#### Échantillonnage

Compléter un programme • Modifier un programme

1. Le programme de la fonction donné ci-contre simule 50 lancers successifs d'une pièce de monnaie équilibrée. La variable *P* compte le nombre de « Pile » obtenus. Compléter ce programme.

2. Quelles modifications faut-il apporter à cette fonction afin qu'elle retourne la fréquence de « Pile » obtenus ?

On définit une nouvelle variable réelle *F*. Puis, après la fin de

la boucle *For*, on introduit l'instruction : « *F=F/50* ». Enfin, on remplace

l'instruction « *return(P)* » par l'instruction « *return(F)* ».

3. Quelles modifications faut-il apporter à cette fonction afin qu'elle simule *n* lancers successifs ?

On définit alors la fonction avec un argument : *def nbre\_pile(n)*.

Puis on remplace l'instruction « *F=F/50* » par « *F=F/n* ».

Enfin, on modifie l'instruction de boucle : « *for i in range(1, n+1)* ».

Boucle bornée « POUR » (For) → p. 18  
Fonctions → p. 26

```
def nbre_pile():
    P=0
    for i in range(.....1,51.....):
        R=randint(0,1)
        P+=R
    return(P)
```

### EXERCICE 89

#### Simulation d'une expérience aléatoire Instruction conditionnelle (If) → p. 14

Comprendre un algorithme

L'algorithme ci-contre simule un jeu opposant Fatima à Maël : ces deux enfants lancent chacun séparément un dé équilibré.

1. Quel est l'affichage en sortie quand :

a. A prend la valeur 6 et B la valeur 5 ?

b. A prend la valeur 3 et B la valeur 4 ?

a. L'affichage en sortie est 6.

b. L'affichage en sortie est 4.

2. Décrire la règle du jeu que simule cet algorithme.

Fatima et Maël lancent chacun un dé. Le résultat affiché est celui du dé qui a le plus grand numéro.

Variables	<i>A</i> et <i>B</i> sont des entiers
Initialisation	<i>A</i> prend pour valeur un entier aléatoire compris entre 1 et 6
Traitement et sortie	<i>B</i> prend pour valeur un entier aléatoire compris entre 1 et 6 Si <i>A</i> > <i>B</i> Alors Afficher <i>A</i> Sinon Afficher <i>B</i> Fin Si

### EXERCICE 90

#### Échantillonnage

Boucle bornée « POUR » (For) → p. 18

Comprendre un algorithme • Programmer un algorithme

On considère l'algorithme ci-dessous.

1. On saisit en entrée : *N* = 20.

Que fait cet algorithme ?

Cet algorithme simule 20 fois une expérience aléatoire consistant à choisir au hasard un entier entre 1 et 6, et il affiche les 20 résultats obtenus.

2. Programmer cet algorithme. [www.](#)

Variables	<i>N</i> , <i>I</i> et <i>R</i> sont des entiers
Entrée	Demander la valeur de <i>N</i>
Traitement et sortie	Pour <i>I</i> variant de 1 à <i>N</i> faire Affecter à <i>R</i> un entier aléatoire compris entre 1 et 6 Afficher <i>R</i> Fin Pour

### EXERCICE 91

Simulation d'une expérience aléatoire Instruction conditionnelle (If) → p. 14

Compléter un algorithme • Programmer un algorithme

Une boîte contient 5 billes rouges, 8 billes noires et 10 billes jaunes. On tire au hasard une bille dans la boîte et on note sa couleur.

1. Compléter l'algorithme ci-contre afin de simuler cette expérience aléatoire.

2. Programmer cet algorithme. [www.](#)

Variable

*R* est un entier

Affecter à *R* un entier aléatoire compris entre 1 et ... 23

Si *R* ≤ 5

Alors Afficher « la bille tirée est rouge »

Sinon Si ..... *R* ≤ 13

Alors Afficher ..... « la bille tirée est noire »

Sinon Afficher ..... « la bille tirée est jaune »

Fin Si

Fin Si

### EXERCICE 92

#### Échantillonnage

Comprendre un programme • Modifier un programme

Le programme ci-contre, écrit en langage Python, simule la répétition d'une expérience aléatoire.

1. On suppose que la variable *R* prend successivement les valeurs : 3, 5, 1, 4, 2, 3, 2, 5, 1, 2.

Quelle valeur est affichée ?

Puisqu'on obtient deux fois le chiffre 5, la valeur affichée en sortie pour *S* est 2.

2. Préciser ce que fait ce programme.

Ce programme simule 10 fois l'expérience aléatoire qui consiste à tirer au hasard un entier entre 1 et 5, et il permet de compter le nombre de fois où on obtient 5. Le nombre de 5 obtenus est affiché en sortie.

3. Donner une expérience aléatoire que peut simuler ce programme.

Par exemple, le tirage au hasard d'un jeton dans une boîte contenant cinq jetons indiscernables numérotés 1, 2, 3, 4 et 5.

4. Quelles modifications doit-on apporter à ce programme pour afficher la fréquence de « 5 » obtenus ?

On introduit une nouvelle variable réelle *F*. Puis, après la boucle *For*, on insère l'instruction « *F=S/10* ». Enfin, on remplace l'affichage par « *print(F)* ».

```
S=0
for i in range(1,11):
    R=randint(1,5)
    if R==5:
        S=S+1
print(S)
```

### EXERCICE 93

#### Simulation d'une expérience

Algorithme – Variables – Affectation → p. 3

aléatoire

Analyser une situation • Écrire un programme

On lance deux dés équilibrés. On calcule le produit des numéros apparus sur la face supérieure de chacun des dés. Écrire un programme qui simule cette expérience aléatoire.

Python

```
A=randint(1,6)
B=randint(1,6)
P=A*B
print(P)
```

Scilab

```
A=tirage_entier(1,1,6)
B=tirage_entier(1,1,6)
P=A*B
disp(P)
```

Xcas

```
A:=hasard(1,1,6);
B:=hasard(1,1,6);
P:=A*B;
afficher(P);
```



**EXERCICE 94**

Échantillonnage

Boucle non bornée « TANT QUE » (While) → p. 22

Compléter un programme • Modifier un programme

On lance un dé équilibré à quatre faces jusqu'à ce que l'on obtienne le numéro 4 sur sa face supérieure.

On veut écrire un programme qui donne en sortie le nombre de lancers de dé nécessaires afin d'obtenir 4 (on dit que c'est le « temps d'attente » pour obtenir le 4).

1. Compléter le programme ci-dessous, écrit en langage Python.

2. Modifier ce programme afin que cette épreuve soit répétée 200 fois et que l'on affiche en sortie la moyenne de ces 200 temps d'attente pour obtenir le numéro 4.

On introduit trois nouvelles variables  $S$  et  $i$ , entiers, et  $M$  flottant. On initialise la variable  $S$  à 0.

```
S=0
for i in range(1,201):
    N=1
    R=randint(1,4)
    while R!=4:
        N=N+1
        R=randint(1,4)
    S=S+N
M=S/200
print(M)
```

Puis on affiche la moyenne  $M$  des temps d'attente en sortie.

```
N=1
R=randint(1,4)
while R!=4:
    N=N+1
    R=randint(1,4)
print(N)
```

**EXERCICE 95**

Prise de décision sur échantillon

Instruction conditionnelle (If) → p. 14  
Fonctions → p. 26

Compléter un programme

On veut savoir si une pièce de monnaie est truquée. Pour cela, on la lance  $N$  fois (avec  $N \geq 25$ ) et on s'intéresse à la fréquence d'apparition du côté « Pile » de la pièce.

1. On fait l'hypothèse que la pièce n'est pas truquée. Déterminer un intervalle de fluctuation au seuil de 95 % de la fréquence de « Pile » dans les échantillons de taille  $N$ .

Un intervalle de fluctuation au seuil de 95 % de la fréquence de « Pile » est  $[0,5 - \frac{1}{\sqrt{N}}, 0,5 + \frac{1}{\sqrt{N}}]$ .

2. Sur  $N$  lancers, le nombre de « Pile » obtenus est  $n$ . On souhaite que la fonction `piece_truquée`, dont on donne le programme en langage Python ci-dessous, donne une règle au seuil de 95 % pour décider si une pièce de monnaie est truquée ou non.

Compléter ce programme.

(and est l'écriture dans le langage Python du connecteur « et » en raisonnement logique.)

```
def piece_truquée(n,N)
    a=0.5-1/sqrt(N)
    b=0.5+1/sqrt(N)
    f=n/N
    if f>=a and f<=b:
        return(".....pièce...non...truquée.....")
    else:
        return(".....pièce...truquée.....")
```

**Programmation avec Scilab et Xcas**

	Scilab	Xcas
Créer un nouveau programme	Quand on ouvre le logiciel, une fenêtre appelée console apparaît à l'écran. On saisit le programme dans l'éditeur.	L'éditeur apparaît à l'écran au démarrage. Dans le menu <b>Prg</b> , on choisit <b>Nouveau programme</b> .
Saisir A	Ouvrir l'éditeur en cliquant sur	Cliquer sur <b>Annuler</b> pour un programme, ou sur <b>OK</b> pour une fonction
Afficher A	disp(A) ou avec le module « lycée » : <b>afficher(A)</b>	<b>print(A)</b> ou <b>afficher(A)</b>
Afficher un texte, par exemple « oui »	disp("oui")	<b>print("oui")</b> ou <b>afficher("oui")</b>
Affecter à B la valeur de A	B=A	<b>B:=A</b>
Écrire des commentaires	On écrit les commentaires sur une ligne, précédés de //	
Tester si $A = B$ / si $A \neq B$	$A==B$	$A <> B$
Tester si $A \geq B$ / si $A \leq B$	$A>=B$	$A <= B$
$A$ et $B$ / $A$ ou $B$	$A&B$	$A B$
Si {condition C} Alors {instructions A} Sinon {instructions B} Fin Si	if {condition C} then {instructions A} else {instructions B} end	si {condition C} alors {instructions A} sinon {instructions B} fsi
Pour $i$ variant de 1 à $n$ {instructions} Fin Pour	for i=1:n {instructions} end	pour k de 1 jusque n faire {instructions} fpour
Tant que {condition C} {instructions} Fin Tant que	while {condition C} {instructions} end	tantque {condition C} faire {instructions} ftantque
Définition d'une fonction $f$	function y=f(a,b,c) y=.... endfunction	<b>f(a,b,c):={</b> retourne ....};
Racine carrée de $x$	$\text{sqrt}(x)$	$\text{sqrt}(x)$
$x$ à la puissance $n$	$x^n$	
$\pi$	$\%pi$	$\pi$
Reste de la division de $a$ par $b$	<b>reste(a,b)</b>	<b>irem(a,b)</b>
Nombre décimal aléatoire compris entre $a$ et $b$	<b>tirage_reel(1,a,b)</b>	<b>hasard(a,b)</b>
Entier aléatoire compris entre $a$ et $b$	<b>tirage_entier(1,a,b)</b>	<b>hasard(1,a,b)</b>
Longueur d'une chaîne de caractères $C$	<b>length(C)</b>	<b>dim(C)</b>
Extraire un caractère d'une chaîne de caractères $C$	<b>part(C,k)</b> renvoie le $k^{\text{e}}$ élément de $C$	$C[k]$ renvoie le $(k+1)^{\text{e}}$ élément de $C$
Concaténation de deux chaînes de caractères	"abcd" + "efg" donne "abcdefg"	
Exécuter un programme	Cliquer sur  ou sur  pour enregistrer, puis exécuter	Cliquer sur <b>OK</b>