

1 Déterminer le type d'une variable à partir d'un contexte

→ Cours 1 p. 18

On souhaite écrire un programme dans lequel apparaissent des variables **ville**, **pluviometrie** et **ensoleillement** donnant respectivement le nom de la ville considérée, sa pluviométrie (en cm par m² arrondi à 0,1) et son nombre de jours d'ensoleillement de l'année en cours. Donner un type possible pour chacune de ces variables.

Solution

La variable **ville** prend des mots pour valeur, elle est donc de type chaîne de caractères. 1

La variable **pluviometrie** prend pour valeur des nombres décimaux, non entiers a priori, elle est donc de type flottant. 2

La variable **ensoleillement** prend pour valeur des nombres de jours c'est-à-dire des nombres entiers, elle est donc de type entier. 2

Conseils & Méthodes

1 On identifie les variables qui ne prennent pas des valeurs numériques : elles sont de type chaîne de caractères.

2 Pour les variables prenant des valeurs numériques, on identifie si ces valeurs sont nécessairement entières ou non. Sinon, elles sont de type flottant.

À vous de jouer !

1 Dans un programme permettant de tester si un triangle est isocèle, équilatéral, rectangle ou quelconque, apparaissent 4 variables : **nom**, **longueur 1**, **longueur 2** et **longueur 3** prenant respectivement pour valeurs le nom du triangle (par exemple ABC) et les longueurs de ses trois côtés. De quel type est chacune de ces 4 variables ?

2 Dans un programme permettant de calculer la moyenne de plusieurs notes dans une certaine matière, apparaissent (entre autres) trois variables : **matière**, **n** et **moyenne** prenant respectivement pour valeurs le nom de la matière, le nombre de notes et la moyenne. De quel type est chacune de ces 3 variables ?

→ Exercices 35 à 36 p. 30

2 Déterminer les valeurs prises par les variables d'un algorithme

→ Cours 2 p. 18

Quelles sont les valeurs prises par les variables **x** et **y** à la fin de cet algorithme ?

Solution

On dresse le tableau suivant. 1

x	5
y	8

On calcule la nouvelle valeur prise par **x** en utilisant les valeurs « actuelles » de **x** et **y** à savoir 5 et 8. 2

x	5	$4 \times 5 + 3 \times 8 = 44$
y	8	

On calcule la nouvelle valeur prise par **y** en utilisant les valeurs « actuelles » de **x** et **y** à savoir 44 et 8. 2

x	5	44
y	8	$2 \times 44 - 7 \times 8 = 32$

À la fin de l'algorithme, **x** a pour valeur 44 et **y** a pour valeur 32.

```
x ← 5
y ← 8
x ← 4 × x + 3 × y
y ← 2 × x - 7 × y
```

Conseils & Méthodes

1 On dresse un tableau ayant, en en-tête, des lignes le nom des variables utilisées et contenant leur valeur à la première affectation (initialisation).

2 Lors d'une nouvelle affectation, on écrit la nouvelle valeur de la variable en utilisant les valeurs actuelles des variables utilisées (les plus à droite dans le tableau).

À vous de jouer !

3 Donner les valeurs prises par les différentes variables à la fin de cet algorithme.

```
a ← 3
b ← 15
b ← b/a
a ← a2 + b2
b ← a × b
```

4 Donner les valeurs prises par les différentes variables à la fin de ce programme.

```
x = 1
y = 2
z = 3
x = x*y*z
z = x + y + z
y = x - y - z
```

→ Exercices 37 p. 30 à 42 p. 31

3 Comprendre une instruction conditionnelle

→ Cours 3 p. 19

On considère l'algorithme ci-contre.

1. Qu'affiche cet algorithme si l'utilisateur saisit pour la variable x ?
2. Qu'affiche cet algorithme si l'utilisateur saisit la valeur 12547 pour la variable x ?

Solution

1. Comme $8 \leq 100$, la condition $x \leq 100$ est vérifiée ¹ donc l'algorithme affiche **Quel manque d'originalité !** ²
L'exécution de l'algorithme se poursuit ensuite après **Fin si** donc l'algorithme affiche **Merci d'avoir participé.** ⁴
2. Comme $12547 > 100$, la condition $x \leq 100$ n'est pas vérifiée ¹ Donc l'algorithme affiche **Bravo !** ³
L'exécution de l'algorithme se poursuit ensuite après **Fin si** donc l'algorithme affiche **Merci d'avoir participé.** ⁴

```
Afficher "Saisir un nombre entier"
x ← Valeur saisie
Si x ≤ 100
    Afficher "Quel manque d'originalité !"
Sinon
    Afficher "Bravo !"
Fin si
Afficher "Merci d'avoir participé."
```

Conseils & Méthodes

- 1 On vérifie que la condition est vérifiée.
- 2 Si la condition est vérifiée, on exécute le bloc d'instructions entre **Si** et **Sinon**.
- 3 Si la condition n'est pas vérifiée, on exécute le bloc d'instructions de **Sinon** à **Fin si**.
- 4 Les instructions conditionnelles passées, on exécute la suite de l'algorithme.

À vous de jouer !

- 5 Qu'affiche l'algorithme suivant si $x=5$? si $x=15$?

```
x ← Valeur saisie
Si 2x - 20 ≥ 0
    Afficher "2x - 20 est positif"
Sinon
    Afficher "2x - 20 n'est pas positif"
Fin si
```

- 6 On considère l'algorithme suivant où la variable **annee** a pour valeur l'année en cours.

Qu'affichait cet algorithme
a) en 2012 ?
b) en 2014 ?

```
Si annee/4 est entier
    Afficher "Année bissextile"
Fin si
```

→ Exercices 43 à 47 p. 31

4 Écrire une instruction conditionnelle

→ Cours 3 p. 19

Dans un cinéma, la place adulte (à partir de 18 ans) coûte 11 euros et la place enfant (moins de 18 ans) coûte 7 euros. Recopier et compléter l'algorithme ci-contre afin qu'il affiche le tarif que doit payer un utilisateur.

Solution

- Le tarif est différent suivant **si** $\text{age} \geq 18$ ou $\text{age} < 18$.
On peut choisir comme condition à tester $\text{age} \geq 18$ (on aurait pu prendre $\text{age} < 18$). ¹
- Si $\text{age} \geq 18$ alors le tarif est de 11 euros, on affiche donc 11 euros. ²
 - Sinon (si $\text{age} < 18$) alors le tarif est de 7 euros, on affiche donc 7 euros. ³

```
age ← Valeur saisie
Si age ≥ 18
    Afficher "11 euros"
Sinon
    Afficher "7 euros"
Fin si
```

Conseils & Méthodes

- 1 On identifie la condition à tester.
- 2 On identifie les instructions à exécuter si la condition est vérifiée.
- 3 On identifie les instructions à exécuter sinon, c'est-à-dire si la condition n'est pas vérifiée.

```
age ← Valeur saisie
Si...
```

À vous de jouer !

- 7 Soit une fonction f définie par $f(x) = x^2$ si $x \geq 0$ et $f(x) = -x^3$ si $x < 0$.
Compléter l'algorithme afin qu'il calcule $f(x)$.

```
x ← Valeur saisie
Si...
```

- 8 Écrire un algorithme demandant à l'utilisateur de saisir une valeur pour une variable **mdp** et affichant **mot de passe non sécurisé** si l'utilisateur a saisi 1234 pour **mdp** et **mot de passe sécurisé** sinon.

→ Exercices 48 à 50 p. 31

5 Comprendre une boucle bornée

→ Cours 4 p. 20

Quelle est la valeur de la variable u à la fin de cet algorithme ?

Solution

On dresse le tableau suivant 1 2.

	Initialisation	$i = 0$	$i = 1$	$i = 2$
u	12			

À chaque passage dans la boucle, on calcule successivement les valeurs des différentes variables. 3

	Initialisation	$i = 0$	$i = 1$	$i = 2$
u	12	$2 \times 12 + 0 - 4 = 20$	$2 \times 20 + 1 - 4 = 37$	$2 \times 37 + 2 - 4 = 72$

La valeur de u à la fin de l'algorithme est donc 72.

```
u ← 12
Pour i allant de 0 à 2
  u ← 2u + i - 4
Fin pour
```

Conseils & Méthodes

- 1 On dresse un tableau ayant, en en-tête, des lignes les variables utilisées autres que le compteur, une colonne pour l'initialisation de ces variables et des colonnes appelées $i = \dots$ pour chaque passage dans la boucle pour.
- 2 On complète la colonne d'initialisation.
- 3 On calcule au fur et à mesure des passages dans la boucle les différentes valeurs des variables.

À vous de jouer !

9 Quelles sont les valeurs prises par la variable r pendant l'exécution de cet algorithme ?

```
Pour i allant de 5 à 7
  r ← 2 × i² + 5
Fin pour
```

10 Quelle est la valeur de la variable b en fin de programme PYTHON ci-contre ?

```
a=0
b=0
for i in range(1,5):
  a=a+2
  b=b+a
```

→ Exercices 51 p. 31 à 54 p. 32

6 Écrire une boucle bornée

→ Cours 4 p. 20

Écrire un algorithme qui affiche les nombres pairs de 2 à 1000.

Solution

On constate qu'il va y avoir un affichage (et éventuellement un calcul) qui se répète 500 fois (afficher successivement 2, 4, 6, 8, ..., 1000) : 1 on peut donc penser à l'instruction **Pour i allant de 1 à 500.** 2

Quand $i=1$ on affiche 2, quand $i=2$ on affiche 4, ..., quand $i=500$ on affiche 1000 : dans tous les cas, il s'agit d'afficher le nombre $2 \times i$ c'est-à-dire d'exécuter l'instruction **Afficher $2 \times i$** à chaque passage dans la boucle. 3

En conclusion, l'algorithme ci-contre répond au problème posé (mais il y a d'autres possibilités).

```
Pour i allant de 1 à 500
  Afficher 2×i
Fin pour
```

Conseils & Méthodes

- 1 On identifie que des instructions vont se répéter un nombre déterminé de fois.
- 2 On en déduit une instruction du type **Pour i allant de... à...**
- 3 On écrit ces instructions (qui se répètent) éventuellement en utilisant le compteur i .

À vous de jouer !

11 Écrire un algorithme qui affiche tous les nombres entiers de 9 à 784.

12 Écrire un algorithme affichant tous les nombres impairs de 1 à 999.

Coup de pouce

• $1 = 1 + 2 \times 0$ • $3 = 1 + 2 \times 1$ • $5 = 1 + 2 \times 2$ etc

13 Recopier et compléter l'algorithme suivant pour qu'il demande 10 fois à un utilisateur de saisir une valeur pour une variable x et ajoute à chaque fois cette valeur à celle de la variable somme.

```
somme ← 0
Pour i allant de... à...
  ... ← Valeur demandée
  somme ← ...
Fin pour
```

→ Exercices 55 à 57 p. 32

7 Comprendre une boucle non bornée

→ Cours 5 p. 20

On considère l'algorithme ci-contre.
Quelle est la valeur affichée par l'algorithme ?

```
i ← 0
u ← 12
Tant que u < 10000
  i ← i+1
  u ← 10×u - 4×i+1
Fin tant que
Afficher i
```

Solution

On dresse le tableau suivant : 1 et 2

	Initialisation
i	0
u	12
Condition u < 10 000	

On vérifie si la condition $u < 10000$ est vérifiée et, comme c'est le cas ici, on exécute les instructions dans la boucle. 3

	Initialisation	
i	0	$0 + 1 = 1$
u	12	$10 \times 12 - 4 \times 1 + 1 = 117$
Condition u < 10000	Vérifiée	

On répète l'étape précédente jusqu'à ce que la condition $u < 10000$ ne soit plus vérifiée. 3

	Initialisation			
i	0	1	2	3
u	12	117	1163	11 619
Condition u < 10000	Vérifiée	Vérifiée	Vérifiée	Non vérifiée

Quand u vaut 11 619, la condition $u < 10 000$ n'est plus vérifiée donc on exécute la fin de l'algorithme c'est-à-dire qu'on affiche la valeur de i : l'algorithme affiche 3. 4

Conseils & Méthodes

- 1 On dresse un tableau ayant en en-tête des lignes les variables utilisées ainsi que la condition (en dernière ligne) et une colonne pour l'initialisation.
- 2 On complète la colonne d'initialisation.
- 3 On teste si la condition est vérifiée et, si oui, on exécute les commandes puis on reteste si la condition est vérifiée, etc.
- 4 Quand la condition n'est plus vérifiée, on exécute la suite de l'algorithme.

À vous de jouer !

14 Quelle est la valeur affichée par l'algorithme suivant ?

```
u ← 5
n ← 0
Tant que u ≠ 656
  u ← 5 × u + 1
  n ← n + 1
Fin tant que
Afficher n
```

15 Quelle est la valeur affichée par l'algorithme suivant ?

```
u ← 5
v ← 12
i ← 1
Tant que u < v
  u ← 3 × u + 2
  v ← 2 × v + 3
  i ← i + 1
Fin tant que
Afficher i
```

16 Quelles sont les valeurs prises par chacune des variables à la fin du programme python suivant ?

```
a=78
b=4
q=0
r=a
while r>=b:
  r=r-b
  q=q+1
```

17 1. Faire tourner l'algorithme suivant sur quelques passages dans la boucle.

2. Décrire par une phrase ce que fait cet algorithme (on ne demande pas son affichage).

3. En déduire l'affichage de fin d'algorithme.

```
x ← 0
Tant que x² < 1000
  x ← x + 1
Fin tant que
Afficher x
```

→ Exercices 58 à 60 p. 32

8 Écrire une boucle non bornée

→ Cours 5 p. 20

Compléter le bloc d'instructions suivant pour qu'après avoir généré un nombre entier aléatoire x entre 1 et 6, il demande à l'utilisateur de saisir des valeurs (stockées successivement dans la variable a) jusqu'à trouver x .

Solution

La demande à l'utilisateur de saisir une valeur se répète mais on ne sait pas combien de fois donc on va utiliser une boucle **Tant que**. ¹

Dans chaque passage dans la boucle, l'utilisateur doit saisir une valeur pour a donc l'instruction à répéter est $a \leftarrow$ Valeur saisie. ²

La condition mettant fin à cette répétition est $a = x$ donc il y a répétition **Tant que $a \neq x$** . ³

Le bloc d'instructions ci-dessous répond donc au problème posé.

```
x ← entier aléatoire entre 1 et 6
a ← 0
Tant que a ≠ x
    a ← Valeur saisie
Fin tant que
```

► **Remarque** On a initialisé a de sorte d'assurer un premier passage dans la boucle mais on aurait pu procéder autrement, par exemple, comme le programme ci-contre.

```
x ← entier aléatoire entre 1 et 6
a ← 0
```

Conseils & Méthodes

¹ On identifie que des instructions vont se répéter un nombre non déterminé de fois.

² On écrit ces instructions (qui se répètent).

³ On identifie la condition qui doit être vérifiée pour que les instructions continuent à se répéter et on en déduit une instruction du type

Tant que Condition.

Il peut arriver qu'il soit plus simple de traiter l'étape ³ avant l'étape ².

```
x ← entier aléatoire entre 1 et 6
a ← Valeur saisie
Tant que a ≠ x
    a ← Valeur saisie
Fin tant que
```

À vous de jouer !

18 On considère le nombre $a = 5$ auquel on applique le programme de calcul on multiplie par -2 et on ajoute 1 , le résultat donnant la nouvelle valeur prise par a .

Écrire un algorithme affichant la première valeur supérieure à 20 prise par a quand on répète ces instructions.

19 Écrire un algorithme calculant et affichant les puissances de 5 successives jusqu'à ce que l'une d'elle dépasse $1\,000\,000$.

👍 **Coup de pouce** On pourra se demander quelle est la plus petite puissance de 5 (pour l'initialisation) et comment passer d'une puissance de 5 à la suivante.

20 1. Écrire un programme :

Étape 1 : demandant à l'utilisateur de saisir un nombre entier a strictement positif.

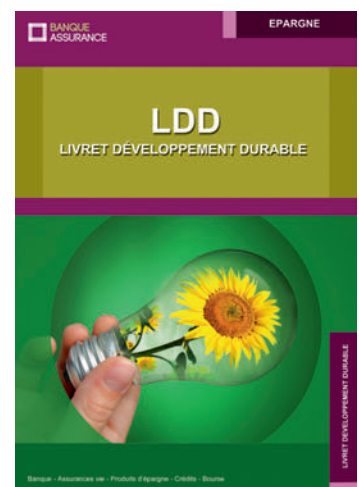
Étape 2 : calculant a^a et affectant la valeur obtenue à a puis affichant le résultat obtenu.

Étape 3 : reprenant l'étape précédente si la nouvelle valeur de a est plus petite que 10^{100} , s'arrêtant sinon.

2. Que se passe-t-il si l'utilisateur saisit 1 comme valeur de a au départ ?

21 Une somme de $1\,000$ euros est placée sur un livret bancaire rémunéré à 2% d'intérêt en 2019. Compléter l'algorithme suivant afin qu'il détermine à partir de quelle année la somme d'argent sur le livret bancaire sera supérieure à $2\,000$ euros.

```
somme ← 1000
annee ← 2019
Tant que...
    somme ← ...
    annee ← ...
Fin tant que
Afficher...
```



→ Exercices 61 à 62 p. 32

9 Utiliser une fonction simple

→ Cours 6 p. 21

Donner l'affichage réalisé par l'algorithme ci-contre.

Solution

Dans l'algorithme principal, on identifie l'appel de la fonction **calculproduit** : c'est **calculproduit(a,b)** qui est équivalent à **calculproduit(5,8)** (car **a=5** et **b=8**). 1

On exécute donc la fonction où **x** a pour valeur **5** et **y** a pour valeur **8** (on respecte l'ordre des paramètres !) : 2
on a donc **produit = 5 × 8 = 40** (dans la fonction).

La fonction retourne la valeur de la variable **produit** donc **40**. 3

Ainsi, dans l'algorithme principal, on peut remplacer **calculproduit(a,b)** par **40**. 4

Il s'agit donc d'exécuter l'instruction **Afficher 40** donc l'algorithme affiche la valeur **40**. 5

```
fonction calculproduit(x,y)
    produit ← x × y
    Retourner produit

a ← 5
b ← 8
Afficher calculproduit(a,b)
```

Conseils & Méthodes

- 1 On repère l'appel d'une fonction.
- 2 On exécute la fonction en donnant aux variables paramètres de la fonction les valeurs données lors de l'appel.
- 3 On détermine la valeur de retour.
- 4 On remplace l'appel de fonction par la valeur de retour.
- 5 On exécute la suite de l'algorithme.

À vous de jouer !

22 Quel est l'affichage de cet algorithme ?

```
fonction bizarre(a,b,c)
    valeur ← 3 × a + 5 × b + c
    Retourner valeur

x ← 0
y ← 1
Afficher bizarre(x,2,y)
```

23 Quelle est la valeur de la variable **a** à la fin de ce programme ?

```
def aire_triangle(base,hauteur) :
    aire=(base*hauteur) / 2
    return aire

a=aire_triangle(4,8)
```

→ Exercices 63 à 64 p. 32

10 Écrire une fonction simple

→ Cours 6 p. 21

Écrire une fonction **calcul_somme** retournant la somme de deux valeurs.

Solution

La fonction a deux paramètres que l'on va appeler **x** et **y**. 1
La valeur de retour est leur somme soit **x + y**. 2
Les deux fonctions ci-contre conviennent. 3

```
fonction calcul_somme(x,y)
    somme ← x + y
    Retourner somme
```

```
fonction calcul_somme(x,y)
    Retourner x + y
```

Conseils & Méthodes

- 1 On identifie les paramètres de la fonction.
- 2 On identifie la valeur de retour et comment l'obtenir.
- 3 On écrit la fonction.

À vous de jouer !

24 Écrire une fonction **moyenne2** prenant comme paramètres deux variables **x** et **y** et retournant la moyenne de leurs deux valeurs.

25 Écrire une fonction **moyenne3** prenant comme paramètres trois variables **x**, **y**, et **z** et retournant la moyenne de leurs trois valeurs.

26 Écrire une fonction **image** prenant en paramètre une variable **x** et retournant l'image de la valeur de **x** par la fonction $x \mapsto 3x^2 - 7x + 4$.

27 Écrire une fonction **hausse20** prenant en paramètre une variable **quantite** et retournant sa valeur augmentée de 20 %.

→ Exercices 65 à 66 p. 32

Apprendre à apprendre



28 Trouver une (ou plusieurs) ressemblance(s) et différence(s) entre les « fonctions » dont parle ce chapitre et les « fonctions » étudiées au collège et dans le chapitre 8 de ce manuel.

29 1. Découper dix cartes identiques dans une feuille cartonnée.

Sur cinq cartes écrire en bleu les instructions \leftarrow , **si**, **pour**, **tant que** et **fonction**.

2. Sur les cinq cartes restantes, écrire en rouge les instructions python correspondantes (sans regarder le cours d'abord puis avec le cours si vous n'y arrivez pas).

3. Battre les cartes et demander à un camarade de les associer.

Questions - Flash



Diapo

Diaporama

Ressource professeur

► **Remarque : PYTHON** dispose d'un module `turtle` permettant de dessiner des figures.

En voici quelques instructions :

- `import turtle` permet de charger le module.
- `t=turtle.Pen()` crée la « tortue ».
- `t.forward(a)` (resp. `t.backward(a)`) fait avancer (resp. reculer) la tortue de a pixels.
- `t.right(b)` (resp. `t.left(b)`) fait se tourner la tortue de b degrés sur sa droite (resp. sa gauche);
- `t.up()` (resp. `t.down()`) lève (resp. baisse) le stylo.

Dans les exercices **30** à **33**, dessiner le motif réalisé par le programme en prenant 1 mm par pixel.

30

```
import turtle
t = turtle.Pen()
t.forward(30)
t.right(90)
t.forward(30)
t.right(90)
t.forward(30)
t.right(90)
t.forward(30)
```

31

```
import turtle
t = turtle.Pen()
t.forward(100)
t.right(135)
t.forward(40)
t.right(45)
t.forward(100)
t.right(135)
t.forward(40)
```

32

```
import turtle
t = turtle.Pen()
t.backward(100)
t.left(90)
t.forward(100)
t.right(90)
t.forward(80)
t.up()
t.backward(80)
t.down()
t.left(90)
t.forward(100)
t.right(90)
t.forward(100)
```

33

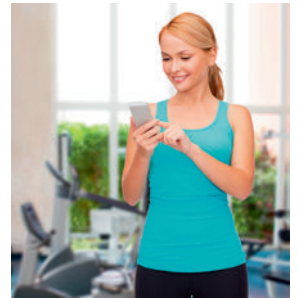
```
import turtle
t = turtle.Pen()
for i in range(1,11):
    t.forward(10)
    t.up()
    t.forward(10)
    t.down()
```

34 À l'aide du module `turtle` de **PYTHON**, dessiner :

- un triangle équilatéral.
- un M majuscule.
- deux segments parallèles et de même longueur (non reliés).

Déterminer le type d'une variable à partir d'un contexte

35 Le programme d'une application sportive comporte, entre autres, trois variables `nom`, `temps` et `total` correspondant au nom de l'utilisateur, au nombre (entier) de minutes passées à faire du sport dans la journée et au nombre de kilocalories dépensées, arrondi à 0,1 kcal.



Donner un type possible pour chaque variable.

36 Un programme permettant de calculer la longueur d'un segment dans un repère orthonormé contient sept variables `nom1`, `nom2`, `abscisse1`, `ordonnee1`, `abscisse2`, `ordonnee2` et `longueur` correspondant au nom des deux extrémités du segment, à leurs coordonnées respectives et à la longueur du segment.

Donner un type possible pour chaque variable.

Déterminer les valeurs prises par les variables d'un algorithme

37 Quelles sont les valeurs des variables en fin de ces algorithmes ?

a)

```
a ← 15
b ← 3 × a² - 7
a ← b - 2
c ← a × b
```

b)

```
x ← 5
y ← 2
x ← -2 × x + 6
y ← 8 × x - 2 × y
```

38 Quelles sont les valeurs des variables en fin de ce programme en **PYTHON** ?

```
x = 12
y = 5
x = 3*x-5
y = 12*x+y
```

39 On considère l'algorithme suivant.

1. Quelle est la valeur prise par la variable x en fin d'algorithme si l'utilisateur saisit 12 pour valeur de x en début d'algorithme ?

```
x ← Valeur saisie
y ← x²
z ← x⁴
x ← y² - z
```

2. Reprendre la question précédente avec d'autres valeurs de x en début d'algorithme. Quelle conjecture peut-on faire ?

3. Démontrer cette conjecture.

40 1. Écrire un algorithme demandant de saisir la longueur du côté d'un carré et affichant son aire.

2. Modifier l'algorithme précédent pour l'adapter à un rectangle.

41 Écrire un algorithme demandant à l'utilisateur de saisir, pour un triangle rectangle :

- les longueurs des deux côtés de l'angle droit et affichant la longueur de son hypoténuse.
- les longueurs d'un côté de l'angle droit et de l'hypoténuse et affichant la longueur du deuxième côté de l'angle droit.

42 Écrire un algorithme demandant à l'utilisateur de saisir une distance parcourue, en km, et une vitesse, en km/h, et affichant le temps, en h, mis pour parcourir cette distance à cette vitesse.

Comprendre une instruction conditionnelle

43 Quelles sont les valeurs des variables en fin de cet algorithme si l'utilisateur saisit :

- 3 pour valeur de x ?
- 2 pour valeur de x ?

```
x ← Valeur saisie
z ← 5
Si x × z > 0
    x ← x - z
Sinon
    z ← 2z
Fin si
z ← z + 2
```

44 1. On considère l'algorithme suivant.

```
Afficher "Taper 1 ou 2."
choix ← Valeur saisie
Si choix = 1
    x ← entier aléatoire entre 1 et 6
Fin si
Si choix = 2
    x ← entier aléatoire entre 1 et 10
Fin si
```

Que fait cet algorithme si l'utilisateur saisit :

- 2 comme valeur pour la variable `choix` ?
- 7 comme valeur pour la variable `choix` ?

2. Expliquer pourquoi l'algorithme précédent est différent de celui ci-dessous.

```
Afficher "Taper 1 ou 2."
choix ← Valeur saisie
Si choix = 1
    x ← entier aléatoire entre 1 et 6
Sinon
    x ← entier aléatoire entre 1 et 10
Fin si
```

45 On considère le bloc d'instructions suivant.

```
Si x × y > 0
    Afficher "x et y sont de signes différents."
Sinon
    Afficher "x et y sont de même signe."
Fin si
```

1. Quel est l'affichage si :

- $x = 3$ et $y = -5$?
- $x = -7$ et $y = -1$?

2. Que faut-il changer pour que le bloc d'instructions soit cohérent ?

46 On considère le bloc d'instructions suivant.

```
Si x > 2 et x ≤ 7
    Afficher "x est dans l'intervalle."
Sinon
    Afficher "x n'est pas dans l'intervalle."
Fin si
```

1. Quel est l'affichage si :

- $x = 5,5$?
- $x = 12$?

2. Ce bloc d'instructions sert à tester si x appartient à un intervalle, lequel ?

47 On considère le bloc d'instructions suivant.

```
Si y ≤ -3 ou y ≥ 36
    Afficher "y n'est pas dans l'intervalle."
Sinon
    Afficher "y est dans l'intervalle."
Fin si
```

1. Quel est l'affichage si :

- $y = 21,7$?
- $y = -36,7$?

2. Ce bloc d'instructions sert à tester si y appartient à un intervalle, lequel ?

Écrire une instruction conditionnelle

48 On considère la fonction f définie par :

$$f(x) = \begin{cases} 2x & \text{si } x > 0 \\ x^3 & \text{si } x \leq 0 \end{cases}$$

Écrire un algorithme ou un programme demandant à l'utilisateur de saisir une valeur de x et affichant son image $f(x)$.

49 Même exercice que le précédent avec :

$$f(x) = \begin{cases} 2x + 3 & \text{si } x \in]-\infty; 1] \\ 5x & \text{si } x \in]1; 2[\\ 14 - 2x & \text{si } x \in [2; +\infty[\end{cases}$$

50 Écrire un algorithme :

- demandant à l'utilisateur de saisir une valeur x .
- donnant le signe de $5x^2 + 3x - 2$ c'est-à-dire si $5x^2 + 3x - 2$ est strictement positif, strictement négatif ou égal à 0.

Comprendre une boucle bornée

51 Qu'affiche l'algorithme suivant ?

```
Pour i allant de 1 à 100
    Afficher 2 × i - 1
Fin pour
```

52 1. Qu'affiche le programme PYTHON suivant ?

```
for x in range(1, 11):
    print(x, "|", 2*x+5)
```

2. Expliquer le lien avec la fonction $x \mapsto 2x + 5$.

3. Modifier le programme pour qu'il permette de tabuler la fonction $x \mapsto 2x^2 + 3x + 5$ entre -1 et 8 avec un pas de 1.

Exercices d'application

- 53** Quelle est la valeur de la variable **somme** à la fin de cet algorithme ?

```
somme ← 0
Pour k allant de 1 à 10
    somme ← somme + k
Fin pour
```

Écrire un algorithme avec une boucle bornée

- 54** Écrire un algorithme ou un programme affichant 1 000 fois le mot « mathématiques ».

- 55** Écrire un algorithme ou un programme permettant d'afficher les images de tous les entiers entre 12 et 55 par la fonction $x \mapsto 3x^2 - 5$.

- 56** Écrire un algorithme ou un programme écrivant tous les triples de 0 à 999 (c'est-à-dire 0 ; 3 ; 6... 996 ; 999).

- 57** Dans une feuille de tableur, on écrit un nombre strictement positif dans la cellule A1.
On écrit ensuite la formule $=A1/2+1/A1$ dans la cellule A2 et on la recopie vers le bas.

1. Écrire un programme ou un algorithme permettant de déterminer la valeur présente dans la cellule A10.
2. Faire tourner l'algorithme à la main avec le nombre de départ de votre choix en A1.
3. De quel nombre « connu » le résultat semble-t-il proche ?

Comprendre un algorithme avec une boucle non bornée

- 58** Quelle est la valeur de **u** en fin de cet algorithme ?

```
u ← 9,5
Tant que u > 0
    u ← 100 - u²
Fin tant que
```

- 59** Quelle est la valeur de **i** en fin de cet algorithme si l'utilisateur choisit 10 pour valeur de **x** ?

```
x ← Valeur saisie
i ← 0
Tant que i < √x
    i ← i + 1
Fin tant que
```

- 60** Expliquer le comportement de ce bloc d'instructions.

```
a ← 0
Tant que a² + 1 > 0
    a ← a - 1
Fin tant que
```

Écrire un algorithme avec une boucle non bornée

- 61** Jeganitha aime bien s'adonner au petit jeu suivant :
- elle part d'un nombre réel positif **j** au hasard,
 - elle le multiplie par 3 et lui ajoute 5,
 - elle recommence avec le nouveau nombre, jusqu'à ce que le résultat soit plus grand que 200.

Écrire un algorithme ou un programme permettant d'afficher le dernier nombre obtenu si elle part du nombre 0.

- 62** On considère la suite logique ci-dessous :
4-5-7-10-14-19-...

Écrire un algorithme ou un programme affichant le premier terme de cette suite logique supérieur ou égal à 100.



Goup de pousse Considérer les différences entre termes consécutifs.

Utiliser une fonction simple

- 63** Quelle est la valeur de retour de la fonction **f** ci-contre si l'on appelle **f(2)** ?

```
fonction f(x)
    y ← 3 × x² + 5
    Retourner y
```

- 64** Quelle est la valeur de retour de la fonction **g** ci-contre si l'on appelle **g(1)** ?

```
fonction g(x)
    y ← x² + 3
    z ← 4 × x + y
    y ← z²
    Retourner z
```

Écrire une fonction simple

- 65** Écrire une fonction retournant l'image d'un réel par
a) $f: x \mapsto -25x + 12$ b) $g: x \mapsto 8x^3 + 5x^2 - 4x + 1$

- 66** En mathématiques, il existe des fonctions de plusieurs variables. Soit la fonction $h: (x; y) \mapsto 3xy - 5x + 2y$
Écrire une fonction retournant l'image d'un couple de réels $(x; y)$ par la fonction **h**.

Calculs et automatismes



- 67** Combien y a-t-il de nombres dans la liste des entiers successifs de 5 à 100 ?

- 68** Calculer $(3x+5y)z$ pour $x = 2$, $y = -1$ et $z = 11$.

- 69** Que doit-on écrire dans les pointillés de la phrase suivante ?

La condition "tant que $i < 35$ " est équivalente à "jusqu'à ce que $i \dots$ ".

70 Quel retour ?


Quel est l'affichage de cet algorithme ?

```

fonction
pythagore(cote1, cote2)
    Retourner  $\sqrt{\text{cote1}^2 + \text{cote2}^2}$ 
a ← 3
b ← 4
c ← pythagore(a,b)
Afficher c
    
```

71 Au tableur !

Dans le tableur, il y a deux fonctions MIN et MAX. Quand on tape `=MIN(5;3)`, la valeur 3 est renvoyée. Écrire deux fonctions MIN et MAX ayant le même usage que celles du tableur.

 **Coup de pouce** C'est-à-dire ayant deux paramètres et retournant la plus grande ou la plus petite des valeurs.

72 Valeur absolue

La valeur absolue d'un nombre x , notée $|x|$, est :

- le nombre lui-même s'il est positif,
- l'opposé du nombre s'il est négatif.

Ainsi, $|4,5| = 4,5$ et $|-3| = -(-3) = 3$, par exemple.

Écrire une fonction de paramètre x qui retourne la valeur absolue de la valeur de x .

73 Hausse et évolution et quantité

1. Écrire une fonction **hausse5** ayant un paramètre **quantite** et retournant la valeur de ce paramètre augmentée de 5 %.
2. Même question pour une fonction **hausse40**.
3. Écrire une fonction **evolution** ayant deux paramètres **quantite** et **taux** et retournant quantite augmentée de taux %.
4. Cette fonction est-elle toujours correcte si **taux** est un nombre négatif ?

74 Pour le 26...

Pour quelle(s) valeur(s) de x l'algorithme affiche-t-il 26 ?

```

x ← Valeur saisie
Si x > 0
    Afficher "Pas intéressant"
Sinon
    y ←  $x^2 + 1$ 
    Afficher y
Fin si
    
```

75 Forfaits au choix

Une entreprise commercialise deux forfaits téléphoniques.

- **Forfait ①** : 1,20 € par heure d'appel.
- **Forfait ②** : illimité pour 20 € par mois.

Écrire un algorithme ou un programme demandant à l'utilisateur de donner son temps d'appel mensuel en heures et affichant le forfait à privilégier.

76 Voyage en train

Pour un trajet de train, un guichet automatique est programmé avec l'algorithme ci-dessous où la variable **age** est l'âge de l'utilisateur.



1. Quel est l'affichage si l'utilisateur a :

- a) 12 ans ?
- b) 36 ans ?
- c) 75 ans ?

```

age ← Valeur saisie
Si age < 27 ou age > 60
    Afficher "15 euros"
Sinon
    Afficher "30 euros"
Fin si
    
```

2. Sous quelles conditions une personne bénéficie-t-elle d'un demi-tarif sur ce trajet de train ?

77 À compléter

Recopier et compléter l'algorithme suivant pour qu'il affiche tous les inverses des entiers de

8 à 20 c'est-à-dire $\frac{1}{8}, \frac{1}{9}$, etc.

```

Pour i allant de... à...
    Afficher...
Fin pour
    
```

78 À partir d'un certain rang...

Écrire un algorithme ou un programme :

- demandant à l'utilisateur de saisir un nombre entier p entre 2 et 10,
- donnant le premier exposant entier i à partir duquel $p^i > 10\,000$.

Par exemple, si l'utilisateur choisit $p = 3$, le début du tableau d'évolution des valeurs des variables est :

p	3	3	3
i	0	1	2
Condition	Vérifiée	Vérifiée	

79 Différence de carrés

1. Écrire un algorithme ou un programme affichant la différence de deux carrés consécutifs de $1^2 - 0^2$ à $100^2 - 99^2$.
2. Le faire tourner « à la main ». Que remarque-t-on ?

80 Pour calculer des aires et des volumes




1. Former des groupes.
2. Lire intégralement l'énoncé du **TP5** p.38, et se répartir toutes les fonctions à écrire entre membres du groupe.
3. Regrouper ensuite les fonctions de tout le groupe dans un seul fichier pour tester le programme.

Exercices d'approfondissement

81 À Monte-Carlo

1. Écrire une fonction **PYTHON** sans paramètre :
 - générant deux flottants x et y au hasard entre -1 et 1 .
 - testant si le point de coordonnées $(x; y)$ est à l'intérieur du cercle de centre O et de rayon 1 dans un repère ortho-normé d'origine O .
 - retournant 1 si c'est le cas et 0 sinon.

 **Coup de pouce** `random.uniform(a,b)` retourne un flottant au hasard entre a et b .

2. Écrire une fonction de paramètre n qui appelle n fois la fonction précédente et retourne $4 \times \frac{p}{n}$ où p est le nombre de fois où le point de coordonnées $(x; y)$ est à l'intérieur du cercle parmi les n appels de la fonction.
3. Lancer plusieurs fois cette dernière fonction avec des valeurs de n élevées (au moins $1\,000\,000$) depuis la console.
4. De quel nombre « connu » semble s'approcher le résultat retourné ?

82 La racine encadrée

Modifier l'algorithme de l'exercice 59 pour qu'il affiche :

- la valeur de \sqrt{x} si x est un carré parfait.
- les deux entiers encadrant \sqrt{x} si x n'est pas un carré parfait.

83 En moyenne !

On souhaite écrire une fonction permettant d'obtenir la moyenne de n notes.

1. Écrire les deux premières lignes d'une fonction moyenne de paramètre n dans laquelle une variable `total` est initialisée à 0 .
2. Écrire une boucle demandant n fois à l'utilisateur de saisir une note et ajoutant à chaque fois cette note au total.
3. Terminer l'écriture de la fonction moyenne retournant la moyenne des n notes saisies.

84 Pour ou Tant que ?

On considère l'algorithme 1 suivant.

Recopier puis compléter l'algorithme 2 pour qu'il fasse exactement la même chose que le précédent.

```
Algorithme 1
u ← 5
Pour i allant de 1 à 5
    u ← 2 × u + 3
Fin pour
```


```
Algorithme 2
i ← 1
u ← 5
Tant que
```

85 Suite de Syracuse

La suite de Syracuse est définie ainsi :

- on part d'un nombre entier strictement positif.
- s'il est pair, on le divise par 2 .
- s'il est impair, on le multiplie par 3 et on lui ajoute 1 .
- on recommence avec le nouveau nombre obtenu.

1. Écrire un programme permettant d'afficher les 100 premiers termes d'une suite de Syracuse (qui dépendra de la valeur de départ).

 **Coup de pouce** L'instruction `if n%2 == 0` permet de tester si un nombre est pair.

2. Exécuter le programme plusieurs fois en changeant le nombre de départ.
Que remarque-t-on ?

86 Un simple échange

Problème ouvert

Écrire un algorithme ou un programme demandant à l'utilisateur de saisir deux nombres a et b et tel que les valeurs de a et b soient échangées.

Par exemple, si l'utilisateur saisit 2 pour valeur de a et 3 pour valeur de b alors $a = 3$ et $b = 2$ en fin de programme.

Vers la 1^{re}



87 Spécialité Maths

La suite de Fibonacci est définie de la manière suivante.

- ① Le 1^{er} et le 2^e termes de la suite sont égaux à 1 .
- ② Tous les termes suivants se calculent en additionnant les deux précédents : le 3^e est $1 + 1 = 2$, le 5^e est $2 + 3 = 5$, le 4^e est $1 + 2 = 3$, etc.

Écrire un programme affichant le $1\,000^{\text{e}}$ terme.

88 Spécialité Maths

Karima a trouvé un algorithme permettant de déterminer le quotient q d'un nombre a par un nombre b . Quand elle l'exécute avec $a = 38$ et $b = 12$, elle obtient successivement : pour $q = 0$, $q \times b = 0 \leq a$, pour $q = 1$, $q \times b = 12 \leq a$, pour $q = 2$, $q \times b = 24 \leq a$, pour $q = 3$, $q \times b = 36 \leq a$, pour $q = 4$, $q \times b = 48 > a$. Donc elle en déduit $q = 4 - 1 = 3$.

- 1. Écrire l'algorithme de Karima en langage naturel.
- 2. Modifier cet algorithme pour le transformer en une fonction de paramètres a et b et qui retourne la valeur q .
- 3. Écrire une fonction de paramètres à déterminer et qui retourne le reste de la division de a par b .

89 STL-ST2S

- Il y a : 4 kcal dans 1 g de glucides, 4 kcal dans 1 g de protéines et 9 kcal dans 1 g de lipides.
- 1. Écrire une fonction `calories100` dont les paramètres sont le nombre de grammes de lipides, de protéines et de glucides par 100 g d'un aliment et renvoyant le nombre de calories au 100 g de cet aliment.
- 2. Écrire le programme principal associé à la fonction précédente demandant à l'utilisateur de saisir le nombre de grammes de lipides, protéines et glucides par 100 g d'un aliment et affichant la valeur renvoyée par la fonction `calories100`.

1 Des conditions avec ET/OU

Logique

TICE

Raisonner

25 min

On considère les programmes **PYTHON** suivants.

Programme 1

```
import random
x=random.randint(1,5)
y=random.randint(1,5)
if x>=2 and y<4:
    print("Vous avez gagné!")
```

Programme 2

```
import random
x=random.randint(1,3)
y=random.randint(1,3)
if x!=3 or y==1:
    print("Vous avez gagné!")
```

1. a) Compléter l'algorithme ci-contre pour qu'il traduise le premier programme.

```
x ← entier aléatoire entre 1 et 5
...
```

b) Le programme 1 affiche-t-il **Vous avez gagné !**

- si $x = 2$ et $y = 3$?
- si $x = 1$ et $y = 5$?
- si $x = 3$ et $y = 4$?

c) Dans un repère, marquer en rouge les points de coordonnées $(x ; y)$ pour lesquels l'affichage est **Vous avez gagné !**

2. a) On rappelle que la condition $x \neq 3$ est vérifiée si la valeur de x est différente de 3 et que la condition $y == 1$ est vérifiée si la valeur de y est égale à 1.

Traduire le deuxième programme en langage algorithmique.

b) Le programme 2 affiche-t-il **Vous avez gagné !**

- si $x = 3$ et $y = 2$?
- si $x = 3$ et $y = 1$?
- si $x = 2$ et $y = 3$?

c) Dans un nouveau repère, marquer en rouge les points de coordonnées $(x ; y)$ pour lesquels l'affichage est **Vous avez gagné !** pour $-5 \leq x \leq 5$ et $-5 \leq y \leq 5$.

2 Découvrir les booléens

TICE

Logique

Raisonner

20 min

1. a) L'inégalité $5 > 3$ est-elle vraie ?

b) Saisir $5 > 3$ dans la console de **PYTHON** et valider. Quel affichage est obtenu ?

c) Proposer une inégalité à écrire dans la console de **PYTHON** qui produise **False** pour affichage. Essayer.

2. a) Écrire le programme ci-contre puis l'exécuter.

b) Ajouter la ligne **print(type(a))** et exécuter le programme.

```
a = 5!=3
print(a)
```

Coup de pouce La commande **type** permet d'afficher le type d'une variable. Ici, la variable **a** est du type **booléen**. Les variables de ce type ne prennent que deux valeurs : **True** ou **False**.

c) Créer une variable **b** de type booléen ayant pour valeur **False**.

3. Quelle est la valeur d'une variable booléenne :

a) **c** définie par **c = (6 > 10) == False** ? Expliquer puis vérifier avec **PYTHON**.

b) **d** définie par **d = (11 != 14) == (4 < 7)** ? Expliquer puis vérifier avec **PYTHON**.

4. On considère le programme ci-contre.

a) Sans le taper, dire ce qu'il va afficher. Vérifier avec **PYTHON**.

b) Le modifier afin qu'il affiche **Super !**

```
e = False
if e:
    print("Super!")
else:
    print("Génial!")
```

3 À propos des groupes sanguins

► **Remarque** Pour réaliser ce TP, les modules **random** et **matplotlib** de **PYTHON** sont nécessaires.

Dans ce TP, tous les programmes commenceront par la ligne **import random**.

On rappelle également que la commande **random.randint(a, b)** de **PYTHON** donne un nombre entier au hasard entre **a** inclus et **b** inclus.

A ► Répartition d'allèles dans une population

Un des gènes qui déterminent le groupe sanguin (présent sur le chromosome 9) existe sous trois versions : l'allèle A, l'allèle B et l'allèle O.

Chaque être humain est porteur de deux allèles (un hérité de chaque parent) qui selon leur dominance/récessivité déterminent son groupe sanguin.

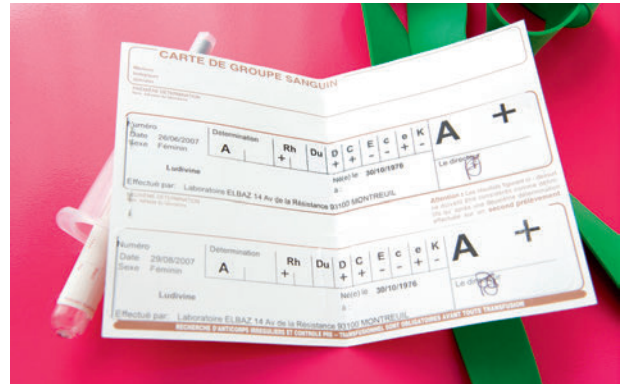
Par exemple un individu porteur des allèles A et B est du groupe AB, un individu porteur des allèles A et O est du groupe A, etc.

On considère une population de 6 individus dans laquelle les 12 allèles (2 par individu) évoqués plus haut sont équirépartis, c'est-à-dire qu'il y a 4 allèles A, 4 allèles B et 4 allèles O.

On souhaiterait observer l'évolution des effectifs de chaque allèle dans les générations suivantes.

Pour cela, recopier le tableau ci-contre à compléter dans les questions suivantes.

Choix du modèle On considère que chaque allèle sera transmis zéro, une ou deux fois à la génération suivante. Modéliser la situation revient donc, pour chacun des allèles, à tirer un nombre entier au hasard entre 0 et 2.



Génération	Effectif de l'allèle A	Effectif de l'allèle B	Effectif de l'allèle O
n° 0	4	4	4
n° 1			
n° 2			
n° 3			
n° 4			

Ce modèle est évidemment très simplifié et discutable ! Normalement, il faudrait, par exemple, appairer les allèles (deux pour chaque individu) qui ne pourraient donc jamais être en effectif total impair.

1. On considère un **programme 1** qui génère un entier au hasard entre 0 et 2 et réalise un affichage comme celui ci contre (obtenu en exécutant 4 fois le programme)

► **Remarque** Le nombre étant aléatoire, on peut obtenir une autre séquence mais toujours avec 0, avec 1 ou avec 2 allèle(s).

Écrire ce **programme 1** avec **PYTHON** (noter les « s » éventuels à « allèle »).

Quatre affichages du programme 1

0 allèle transmis. 1 allèle transmis.
2 allèles transmis. 0 allèle transmis.

2. a) Exécuter quatre fois le **programme 1** afin de simuler l'effectif de l'allèle A à la génération n° 1 à partir des 4 allèles A de la génération n° 0 puis reporter le résultat dans le tableau.

b) Faire de même pour obtenir l'effectif des allèles B et C de la génération n° 1 et reporter ces résultats dans le tableau.

3. Compléter la ligne correspondant à la génération n° 2 à l'aide du **programme 1**.

4. a) Pour gagner du temps, on souhaite écrire le **programme 2** :

- demandant l'effectif d'un allèle à une génération ;
- affichant l'effectif de cet allèle à la génération suivante.

Recopier et compléter les pointillés dans le **programme 2**.

► **Coup de pouce** On rappelle que pour deux entiers **n** et **p**, la ligne **for i in range(n, p)** fait varier **i** de **n** à **p-1** (et non **p**).

Programme 2

```
import random

effectif = input("Effectif de l'allèle?")
effectif = int(effectif)
nouvel_effectif = 0
for i in range(1,...):
    nouvel_effectif = nouvel_effectif + ...
print(...)
```

b) Utiliser ce **programme 2** afin de remplir les lignes suivantes du tableau.

B ► Étude d'effectifs à la génération zéro

Dans cette partie, on va utiliser un programme (constitué notamment d'une partie du **programme 2**) qui :

- demande à l'utilisateur l'effectif de chaque allèle à la génération n° 0 et le nombre de générations souhaité ;
- donne en sortie :
 - l'effectif de chaque allèle après le nombre de générations spécifié ;
 - le graphique de l'évolution de la fréquence de chaque allèle sur toutes ces générations.

1. Recopier le tableau ci-dessous à compléter dans les questions suivantes.

Fréquence (en %) à la génération n° 100 de ...			
Effectif de chaque allèle à la génération n° 0	l'allèle A	l'allèle B	l'allèle O
50			
1 000			
10 000			

Doc Fichiers TICE
Lienmini.fr/maths2-28

2. a) Aller chercher le programme.

b) Exécuter le programme avec un effectif de 50 pour chaque allèle à la génération n° 0 pour 100 générations.

c) Enregistrer puis **fermer la figure générée et noter** l'effectif de chaque allèle après 100 générations puis compléter la première ligne du tableau en calculant les fréquences de chaque allèle.

3. Reprendre la question 2. avec 1 000 comme effectif de chaque allèle à la génération n° 0 pour 100 générations.

4. Reprendre la question 2. avec 10 000 comme effectif de chaque allèle à la génération n° 0 pour 100 générations.

5. Réaliser un compte-rendu numérique des résultats observés dans cette partie B.

Celui-ci contiendra notamment les trois graphiques obtenus et fera le lien entre les effectifs de départ et les fréquences finales des différents allèles.



Raisonner, Calculer



4 Une petite étrangeté

1. Écrire un programme demandant à l'utilisateur de saisir deux nombres réels x et y et testant si leur produit est égal à 6, 3.

2. Tester le programme avec :

a) 1 et 6, 3. b) 2 et 4. c) 3 et 2, 1.

Les affichages obtenus sont-ils cohérents ?

3. Écrire $3 * 2 . 1$ dans la console et valider.

Que remarque-t-on ?

► **Remarque** Les nombres manipulés étant traités en base 2 par l'ordinateur (le fameux binaire), cela induit des approximations problématiques dans les calculs avec des nombres non entiers. C'est pourquoi, d'une manière générale, on évite les tests d'égalité sur les flottants.



5 Pour calculer des aires et des volumes

Le but de ce TP est d'écrire en **PYTHON** un programme permettant de calculer automatiquement des aires et volumes de figures usuelles.

A ► Quelques fonctions

1. Écrire une fonction de première ligne `def aire_carre(c)` retournant l'aire d'un carré dont le côté `a` pour longueur `c`.
2. Écrire une fonction de première ligne `def aire_rectangle(l1,l2)` retournant l'aire d'un rectangle de longueur `l1` et de largeur `l2`.
3. Écrire sur le même modèle les fonctions `aire_triangle`, `aire_disque`, `volume_cube`, `volume_cone` et `volume_boule` en tenant bien compte du nombre de paramètres nécessaires aux calculs de ces aires et volumes.

Coup de pouce On peut tester cette fonction dans la console en rentrant `aire_carre(6)` par exemple.

Coup de pouce Écrire `import math` en début de programme : le nombre π s'obtient alors avec la commande `math.pi`.

B ► Création de menus

1. Écrire une fonction de première ligne `def menu_principal()` :
 - affichant successivement et à la ligne `Pour calculer des aires`, taper 1, puis `Pour calculer des volumes`, taper 2 ;
 - affichant `Votre choix ?` et attendant la saisie d'un nombre de type entier par l'utilisateur pour l'affecter à une variable `choix` ;
 - retournant la valeur de la variable `choix`.
2. Sur le même modèle, écrire une fonction `menu_aire` :
 - affichant successivement et à la ligne : `Pour l'aire d'un carré`, taper 1 puis `Pour l'aire d'un rectangle`, taper 2 puis `Pour l'aire d'un triangle`, taper 3 puis `Pour l'aire d'un disque`, taper 4.
 - affichant `Votre choix ?` et attendant la saisie d'un nombre de type entier par l'utilisateur pour l'affecter à une variable `choix` ;
 - retournant la valeur de la variable `choix`.
3. Écrire sur le même modèle une fonction `menu_volume` proposant de la même manière de calculer le volume d'un cube, d'un cône ou d'une boule.

C ► Programme principal

1. Écrire le début du programme principal suivant puis le compléter.

```
choix1 = menu_principal()
if choix1 == 1:
    choix2 = menu_aire()
    if choix2 == 1:
        c = float(input("Longueur d'un coté ?"))
        print(aire_carre(c))
    if choix2 == ...
```

2. Tester le programme.
3. Pour aller plus loin
 - Ajouter un choix `Pour quitter le programme`, taper 0 dans la fonction `menu_principal` et, à l'aide d'une boucle `Tant que`, faire en sorte de réafficher le menu principal après chaque calcul d'aire ou de volume jusqu'à ce que l'utilisateur saisisse 0.
 - Vous pouvez aussi calculer d'autres aires et volumes (ou périmètres !).

1 Comprendre l'affectation

QCM

Pour les exercices 90 et 91 on considère l'algorithme ci-contre.

```
x ← -1
y ← 5
x ← x - y
y ← x - y
```

90 À la fin de cet algorithme, on a :

- ☐ a $x = -1$ ☐ b $x = -6$ ☐ c $x = -4$

91 À la fin de cet algorithme, on a :

- ☐ a $y = 5$ ☐ b $y = -6$ ☐ c $y = -11$

92 * On considère le programme PYTHON suivant.

```
a=12
b=5
b=a*b
a=3*b
print("La valeur de a est",a)
print("La valeur de b est",b)
```

Qu'affiche ce programme ?

2 Travailler avec des instructions conditionnelles

QCM

93 On considère l'algorithme ci-contre.
Si l'utilisateur saisit 4 comme valeur de x , l'algorithme affiche :

```
x ← Valeur saisie
Si  $-5x + 10 > 0$ 
    Afficher "image positive"
Sinon
    Afficher "image négative"
Fin si
```

- ☐ a 4 ☐ b -10
☐ c image positive ☐ d image négative

94 * On considère la fonction f définie par :

$$f(x) = \begin{cases} 6x - 2 & \text{si } x > 5 \\ 4x + 8 & \text{si } x \leq 5 \end{cases}$$

Écrire un algorithme ou un programme demandant à l'utilisateur de saisir une valeur de x et affichant son image $f(x)$.

3 Comprendre et écrire une boucle

QCM

95 On considère l'algorithme ci-contre.
Cet algorithme affiche :

- ☐ a 2 ☐ b 5
☐ c 24 ☐ d 120

```
i ← 1
Pour j allant de 2 à 5
    i ← i × j
Fin pour
Afficher i
```

Pour les exercices 96 et 97 on considère le bloc d'instructions ci-contre.

```
i ← 0
Tant que s > 0
    s ← s - 100
    i ← i + 1
Fin tant que
Afficher s
Afficher i
```

96 Si la valeur de s est 641 en début de bloc, quelle est la valeur de s affichée en fin de bloc ?

- ☐ a 141 ☐ b 41 ☐ c -59 ☐ d -159

97 Si la valeur de s est 471 en début de bloc, quelle est la valeur de i affichée en fin de bloc ?

- ☐ a 6 ☐ b 5 ☐ c 4 ☐ d 0

98 ** Écrire un algorithme ou un programme qui affiche tous les nombres pairs de 624 à 48 762.

99 ** Écrire un algorithme ou un programme additionnant des nombres entiers aléatoires entre 1 et 10 jusqu'à ce que leur somme soit supérieure à 1 000.

4 Travailler avec les fonctions

QCM

100 Quelle valeur retourne la fonction ci-dessous si l'on appelle `facture(120,12,7)` ?

```
def facture(prix, effectif, reduction):
    total = prix * effectif
    remise = total * reduction / 100
    total = total - remise
    return total
```

- ☐ a 1 440 ☐ b 100,8 ☐ c 1 339,2

101 * Quelle est la valeur de retour de la fonction `affine` suivante si l'on appelle `affine(2, 4)` ?

```
fonction affine(a,b)
    Retourner -b/a
```

102 * Écrire une fonction à deux paramètres nommée `ecart` retournant l'écart entre les deux valeurs prises par les paramètres.

103 ** Écrire une fonction retournant le volume d'un pavé droit (bien choisir les paramètres).

104 ** Écrire une fonction `multiples` à deux paramètres k et n entiers affichant les n premiers multiples non nuls de k .

Nombres et calculs

Diophante d'Alexandrie
(Entre 200 et 500)

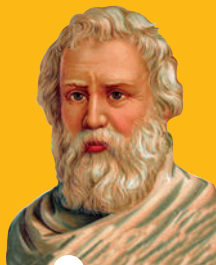


Les Égyptiens, les Mayas et les Romains avaient tous leur propre manière de compter et de faire des calculs. Les nombres servaient notamment au recensement, aux transactions, aux conversions.

Diophante d'Alexandrie, dans son *Arithmétique*, étudie certaines équations dites diophantiennes.

→ [Dicomaths](#) p. 349

Archimède
(vers 287 av. J.-C. – 212 av. J.-C.)



Archimède propose un encadrement de π .

→ [Dicomaths](#) p. 347

Al-Khwarizmi
(vers 780 – vers 850)



Al-Khwarizmi présente des méthodes de résolution de certaines équations du 1^{er} et du 2nd degré.

→ [Dicomaths](#) p. 347

Mon parcours du collège au lycée

Au collège, j'ai utilisé des nombres (décimaux, rationnels, etc.) et des fractions (classiques, irréductibles, décimales, etc.) pour comparer, calculer et résoudre des problèmes. J'ai également découvert la racine carrée, les carrés parfaits entre 1 et 144, les puissances et la notation scientifique. J'ai étudié les notions de divisibilité (division euclidienne, multiples, diviseurs) et de nombres premiers. J'ai utilisé le calcul littéral et la résolution d'équations pour résoudre des problèmes.

En 2^{de}, je vais approfondir mes connaissances sur les divers types de nombres et les ensembles associés afin de résoudre des problèmes arithmétiques. Je vais découvrir la notion d'intervalle réel et apprendre à résoudre des inéquations du 1^{er} degré. Je vais utiliser le calcul littéral et apprendre à résoudre des équations produit, quotient, ou incluant des fonctions de référence.