

Cahier d'algorithmique

Tout le programme en algorithmes

✓ Tous les langages de programmation
(de Scratch à Python, T.I. Basic,
Basic Casio)

✓ Des TP Python
en salle informatique

✓ Des exercices
avec ou sans machine

2^{de}



Cahier Tout le programme en algorithmes **d'algorithmique**

2^{de}

Catherine Divoux
Cédric Gouygou
Nadine Joseph
Gilles Lassus
Léa Magnier
Guillaume Saes

Remerciements : Les auteurs et les Éditions Belin remercient la société Casio pour avoir mis à notre disposition des émulateurs de Graph 35+E et de Graph 90+E pour permettre les captures d'écran dans ce cahier.

Composition et schémas : STDI

Crédits photos :

Couverture: (binaire) : ©iStockphoto/VOLHA RAMANCHUK, (robot) : ©iStockphoto/Cylonphoto
p.12 : ©iStockphoto.com/BrianAJackson, p.13 : ©iStockphoto.com/VictorHuang
p.15 : ©iStockphoto.com/nicholashan, p.21 : ©iStockphoto.com/sarah5, p.23 : ©iStockphoto.com/Dreamsofts,
p.25 : ©ESA/Rosetta/NavCam, p.28 : ©iStockphoto.com/Besjunior
p.34 : ©iStockphoto.com/Zapp2Photo, p.51 : ©iStockphoto.com/jauhari1
p.60 : ©iStockphoto.com/adventtr, p.61 : ©iStockphoto.com/Martin Barraud
p.63 : ©iStockphoto.com/damedeeso, p.65 : ©iStockphoto.com/Sashkinw
p.69 : ©iStockphoto.com/Elaineitalia, p.78 : ©iStockphoto.com/jamesbenet

Le code de la propriété intellectuelle n'autorise que « les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » [article L. 122-5]; il autorise également les courtes citations effectuées dans un but d'exemple ou d'illustration. En revanche « toute représentation ou reproduction intégrale ou partielle, sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » [article L. 122-4]. La loi 95-4 du 3 janvier 1994 a confié au C.F.C. (Centre français de l'exploitation du droit de copie, 20, rue des Grands-Augustins, 75006 Paris), l'exclusivité de la gestion du droit de reprographie. Toute photocopie d'œuvres protégées, exécutée sans son accord préalable, constitue une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal.



Sommaire

Comprendre les notions 4

Partie 1 Fonctions

- 1 Étude qualitative de fonctions 11
- 2 Expressions algébriques 17
- 3 Équations - inéquations 23
- 4 Fonctions de référence : fonctions affines et linéaires, fonction carré, fonction inverse 28
- 5 Fonction polynôme de degré 2 36
- 6 Trigonométrie 42

Partie 2 Statistiques et probabilités

- 7 Statistiques descriptives, analyse de données 49
- 8 Échantillonnage 57
- 9 Probabilité 66

Partie 3 Géométrie

- 10 Géométrie repérée 74
- 11 Équations de droites 80
- 12 Vecteurs 86
- 13 Géométrie dans l'espace 93

Focus sur..

- Variable et fonction : mathématiques vs algorithmique 10
- L'usage des connecteurs « ET » et « OU » 48
- Si ... alors ... : implication et instruction conditionnelle 73

Comprendre les notions

1 Algorithme, programmation et variable

Mise en situation

Sam, une régisseuse lumières, travaille pour une grande salle de spectacle. Différents spots lumineux sont placés sur des rails au plafond. Ce plafond est muni d'un repère. Pour commander ces spots, Sam dispose d'une console informatisée dans laquelle elle saisit les coordonnées des différents spots. Pour un spectacle de danse, parmi les consignes, on lui demande de positionner un spot rose R entre deux spots A et B de lumière jaune à égale distance de chacun de ces spots. Sam doit élaborer un programme qui prend en compte les

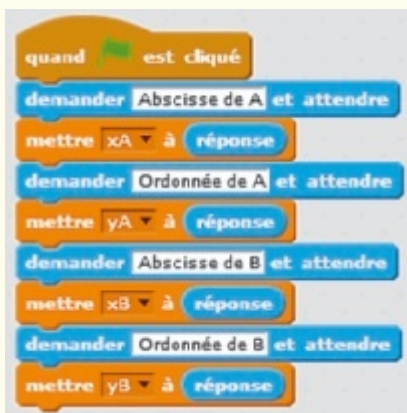
coordonnées de A et B , qu'elle voit sur le repère du plafond, et qui positionne le spot rose R exactement au milieu du segment $[AB]$.

On peut modéliser cette situation en utilisant le logiciel Scratch et un programme en deux parties :

- Lorsque le bouton « Drapeau vert » est pressé, le programme demande l'abscisse et l'ordonnée du spot A puis celles du spot B .

- Ensuite, lorsque la barre espace est pressée, le logiciel affiche les deux spots A et B , ainsi que le spot R situé au milieu segment $[AB]$.

Voici un extrait de ce programme :



Définition De l'algorithme à la programmation

Un algorithme est une suite finie d'instructions simples à réaliser dans un ordre donné de manière à atteindre un (ou des) objectif(s) fixé(s) au préalable.

Il peut être ensuite traduit dans un **langage de programmation** qui peut être un langage « par blocs » (comme le logiciel Scratch) ou un langage textuel. Il deviendra alors un **programme** qui sera exécuté par une machine.

1 Citez des noms de langages de programmation dont vous avez déjà eu connaissance.

Scratch, Casio, etc.

Définition La notion de variable en informatique

En informatique, une **variable** est considérée comme une « boîte » portant une étiquette, c'est-à-dire son **nom**, boîte dans laquelle sera **stocké un contenu** : un **nombre entier** ou **flottant** (nombre décimal) ou **une chaîne de caractères**.

On dit qu'on **affecte** un contenu à cette boîte. Le contenu de cette boîte pourra varier au cours de l'exécution du programme. À chaque nouvelle affectation, le contenu précédent de la boîte est « écrasé ».

2 Combien le programme de Sam comporte-t-il de variables ? Quelles sont ces variables ?

Il comporte 6 variables qui sont $x_A, y_A, x_B, y_B, x_R, y_R$.

3 Les affectations de valeurs à ces variables suivent-elles toutes le même protocole ? Justifier.

Les variables x_A, y_A, x_B, y_B ont des affectations entrées par l'utilisateur et ne varient pas au cours de l'exécution du programme. x_R, y_R ont des affectations qui sont calculées par le programme.

Définition Structure d'un algorithme

Un algorithme se décompose en étapes, qui répondent aux questions suivantes.

Quelles sont les variables et leur type ?

Variables

Quelles sont leurs valeurs initiales ?

Initialisation

Que saisit l'utilisateur au départ ?

Entrée

Quelles sont les instructions exécutées par l'algorithme ?

Traitement

Il peut y avoir plusieurs instructions. On peut ensuite soit arrêter l'algorithme, soit demander qu'il affiche certaines valeurs :

Que veut-on afficher à la fin de l'exécution de l'algorithme ?

Sortie

Quand souhaite-t-on arrêter l'algorithme ?

Fin de l'algorithme

4 Voici une écriture possible en langage naturel de l'algorithme correspondant à la saisie des coordonnées de A et B puis au calcul et à l'affichage des coordonnées du milieu R de $[AB]$. Compléter cet algorithme.

Langage naturel

Variables	$x_A, y_A, x_B, y_B, x_R, y_R$
Entrée	Saisir x_A, y_A, x_B, y_B
Traitement	Affecter à x_R la valeur $\frac{x_A + x_B}{2}$
	Affecter à y_R la valeur $\frac{y_A + y_B}{2}$
Sortie	Afficher x_R, y_R

5 Voici la traduction dans trois langages de programmation de l'algorithme précédent. Pour chaque langage, entourer les noms des variables, souligner les étapes d'entrée en bleu, de traitement en rouge, de sortie en vert.



BESOIN D'AIDE ?

Voir les fiches à l'intérieur de la couverture pour les traductions dans les différents langages de programmation.

Première programmation dans un langage textuel

T.I. Basic	Basic Casio	Python
Prgm puis Nouveau	Menu Prgm puis Nouveau	Nouveau Fichier
Input "Abscisse de A=";U	"U=" ? → U ↵	U=float(input("U="))
Input "Ordonnée de A=";V	"V=" ? → V ↵	V=float(input("V="))
Input "Abscisse de B=";W	"W=" ? → W ↵	W=float(input("W="))
Input "Ordonnée de B=";X	"X=" ? → X ↵	X=float(input("X="))
$(U+W)/2 \rightarrow Y$	$(U+W)/2 \rightarrow Y \downarrow$	$Y=(U+W)/2$
$(V+X)/2 \rightarrow Z$	$(V+X)/2 \rightarrow Z \downarrow$	$Z=(V+X)/2$
Disp "Les coordonnées du milieu sont :"	"Les coordonnées du milieu sont :"	print("Les coordonnées du milieu sont :")
Disp Y	Y ▲	print(Y)
Disp Z	Z ▲	print(Z)

2 Notion de fonction au sens algorithmique

Mise en situation

Lors de grands matchs de tennis, la vitesse de la balle frappée est mesurée par des radars. L'australien Samuel Groth détient le record du service le plus rapide, avec une balle flashée à 263,4 km/h (en 2012). Une fois la vitesse relevée en km/h, un entraîneur souhaite connaître la distance parcourue par la balle pour un temps t donné en seconde.

Définition Fonction en algorithmique

En algorithmique, une fonction correspond à une séquence d'instructions réalisant une tâche précise, en utilisant un ou plusieurs **arguments** (les paramètres de la fonction). Cette fonction reçoit **un nom** et l'on précise ses **arguments**. Elle renvoie un résultat. Elle permet de découper le problème étudié en sous-problèmes et d'éviter ainsi la répétition fastidieuse d'instructions.

Une fois une fonction définie, elle peut être « appelée » tout au long de l'exécution de l'algorithme, autant de fois que nécessaire.

1 On souhaite dans un premier temps définir une fonction calculant la distance parcourue par la balle en t secondes. Quels sont les arguments de cette fonction ?

Les arguments de cette fonction sont la vitesse V (km/h) et le temps T (s).

2 L'entraîneur décide de faire calculer la vitesse en m/s, notée A . Compléter le script ci-contre afin que cette fonction renvoie la distance parcourue par la balle (en m) pour un temps T (en s).

Langage naturel

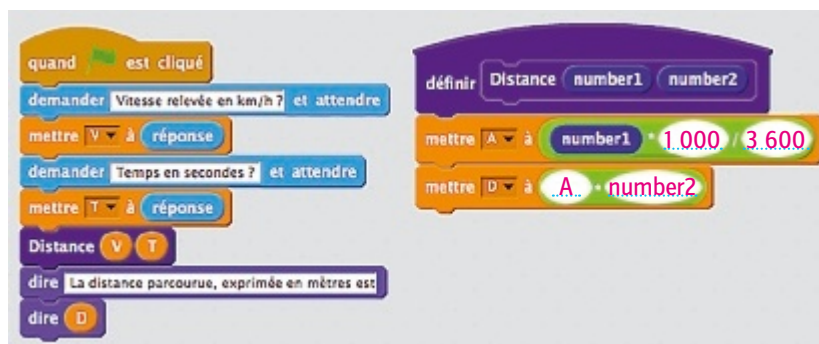
Fonction distance(V , T)

Affecter à A la valeur $\frac{V \times 1000}{3600}$

Affecter à D la valeur $A \times T$

Retourner D

3 Compléter le programme suivant afin de définir cette fonction par un bloc avec le logiciel Scratch.



4 Compléter le script ci-contre écrit en langage Python pour qu'une fois la vitesse V (en km/h) et le temps T (en s) saisis, ce programme calcule et affiche la distance D (en m).

```
def distance(V,T):
    A=V*1000/3600
    D=A*T
    return (D)

V=float(input("Saisir vitesse en km/h"))
T=float(input("Saisir le temps en secondes"))

D=distance(V,T)
print ("La distance parcourue par la balle est, en mètres :")
print (D)
```

3 Instruction conditionnelle

Mise en situation

Une société de location de voitures propose à ses clients le contrat suivant : un forfait de 66 €, auquel s'ajoute 0,25 € par kilomètre parcouru au-delà de 70 km. On souhaite élaborer un algorithme permettant de calculer automatiquement le coût C du contrat, en fonction de la distance parcourue.

Définition Instruction conditionnelle

- **Si une condition donnée est vérifiée,** on effectue la tâche (1).
- **Si une condition donnée n'est pas vérifiée,** on effectue une autre tâche (2).

```

Si <condition>
  alors <tâche 1>
Sinon
  <tâche 2>
FinSi
  
```

Le « sinon » n'est pas obligatoire. Si la condition n'est pas vérifiée, la tâche correspondante n'est pas effectuée et l'algorithme passe à l'instruction suivante.

On peut mettre plusieurs instructions conditionnelles à la suite avant un FinSi.

1 L'algorithme ci-dessous répond-il à la demande initiale ? Si non, le corriger.

Langage naturel

Variables	X et C nombres réels
Entrées	Saisir X
Traitement	Si $(0 < X)$ et $(X < 70)$ Alors C prend la valeur 66 Sinon C prend la valeur $66 + 0,25X$ FinSi
Sortie	Afficher C

```

Variables :    $X$  et  $C$  nombres réels
Entrée :     Saisir  $X$ 
Traitement : Si  $(0 < X)$  et  $(X < 70)$ 
                Alors  $C$  prend la valeur 66
                Sinon si  $X \leq 0$ 
                Alors  $C$  prend la valeur 0
                Sinon  $C$  prend la valeur  $66 + 0,25(X - 70)$ 
                FinSi
Sortie :     Afficher  $C$ 
  
```



BESOIN D'AIDE ?

Que se passe-t-il lorsque l'utilisateur saisit un nombre X négatif ?

2 Voici la traduction dans trois langages de programmation de l'algorithme précédent corrigé. Pour chaque langage, entourer les instructions conditionnelles.

T.I. Basic	Basic Casio	Python
<pre> Prompt X If 0<X and X<70 Then 66 → C Else If X ≤ 0 Then 0 → C Else 66+(X-70)*0.25→C End End Disp « Le coût du contrat en euros est », C </pre>	<pre> "X="?→X↓ If 0<X And X>70↓ Then 66 → C↓ Else↓ If X ≤ 0↓ Then 0 → C↓ Else↓ 66+(X-70)*0.25→C↓ IfEnd↓ IfEnd↓ « Le coût du contrat en euros est : » : C▲ </pre>	<pre> X=float(input("X=")) if 0<X<70 C=66 elif X <= 0 C=0 else C=66+(X-70)*0.25 print(« Le coût du contrat en euros est : ») print(C) </pre>

Comprendre les notions

4 Boucle bornée

Mise en situation

On veut calculer la somme des entiers positifs jusqu'à un nombre N donné.

Définition Boucle bornée

En algorithmique, la **boucle bornée** ou **Boucle Pour** permet de réaliser des instructions pendant un nombre déterminé d'itérations.

Quand le nombre d'itérations demandé est atteint, on sort de la boucle et on exécute les instructions suivantes de l'algorithme.

Pour compter le nombre d'itérations, **il faut définir une variable initialisée à 0 avant la boucle. Cette variable augmentera de 1 à chaque passage dans la boucle.**

1 L'algorithme ci-dessous répond-il à la demande ? Pour le tester, l'exécuter avec $N = 5$ et compléter le tableau suivant.

Pseudo-langage

```
S ← 0
Pour I allant de 1 à N
  S ← S + I
Fin Pour
```

N	5	5	5	5	5	5
I		1	2	3	4	5
S	0	1	3	6	10	15

Cet algorithme calcule bien la somme des entiers de 1 à 5.

2 Voici la traduction dans trois langages de programmation de l'algorithme précédent. Pour chaque langage, entourer la boucle bornée.

T.I. Basic	Basic Casio	Python
O → S Prompt N For(1,1,N) S+I → S End Disp S	O → S "N=" ? → N, I For 1 → I To N Step 1 S+I → S Next S ▲	S=0 N=int(input("N=")) for i in range (1,N+1): S=S+i print (S)

3 Implanter ce programme et donner la plus petite valeur de N pour laquelle on obtient, après exécution de l'algorithme, $S > 200$.

$N = 20$

5 Boucle non bornée

Mise en situation

Au premier janvier 2018, le prix d'une voiture neuve était de 20 000 €. Chaque année, la valeur de cette voiture diminue de 20 %.
On souhaite écrire un algorithme qui calcule le nombre d'années au bout duquel la valeur de cette voiture passe sous le seuil de 2 000 €.

Définition Boucle non bornée

En algorithmique, la **boucle non bornée** ou **Boucle Tant que < condition >** permet de réaliser des instructions **tant que la < condition > est vérifiée**. Quand la condition n'est plus vérifiée, on ne rentre plus dans la boucle et on passe à l'instruction suivante.

1 Calculer la valeur de cette voiture au bout de 2 ans.

Valeur au bout d'un an : $20\,000 \times (1 - 0,2) = 16\,000$.

Valeur au bout de 2 ans : $16\,000 \times (1 - 0,2) = 12\,800$.

2 Compléter l'algorithme ci-dessous pour qu'il calcule le nombre d'années N nécessaire pour que la valeur de cette voiture passe sous le seuil des 2 000 €. On note V la valeur de la voiture.

Langage naturel

Variables N, V
Entrées Saisir V
Traitement Affecter à N la valeur 0
 Tant que $V \geq 2\,000$
 Affecter à V la valeur $0,8V$
 Affecter à N la valeur $N+1$
 FinTantque
Sortie Afficher N

3 Voici la traduction dans trois langages de programmation de l'algorithme précédent. Pour chaque langage, entourer la boucle non bornée.

T.I. Basic	Basic Casio	Python
Prompt V $0 \rightarrow N$ While $V \geq 2000$ $0,8 * V \rightarrow V$ $N+1 \rightarrow N$ End Disp N	"V=?" \rightarrow V \downarrow $0 \rightarrow N$ While $V \geq 2000$ $0,8 * V \rightarrow V$ $N+1 \rightarrow N$ WhileEnd N \blacktriangleleft	$V = \text{float}(\text{input}("V="))$ $N = 0$ while $V \geq 2000$: $V = 0,8 * V$ $N = N + 1$ print (N)

4 Implanter l'un de ces programmes et vérifier qu'il fonctionne.

Variable et fonction : mathématiques vs algorithmique

1 Qu'est-ce qu'une variable ?

■ En mathématiques

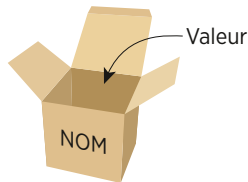
Une variable numérique est une grandeur dont la valeur n'est pas fixée. Elle est souvent désignée par un **symbole** de type lettre (par exemple, x , t , θ , n ...). Cette grandeur appartient à un ensemble bien défini. À cette grandeur peuvent être associées des expressions algébriques dont la valeur dépend de celle que prend la variable.

Exemple : Pour tout réel x , l'expression $A(x) = x^2 - 2x + 1$ est définie en fonction de la variable x . Pour $x = 3$, $A(3) = 3^2 - 2 \times 3 + 1 = 4$.

■ En algorithmique

Une variable est une information stockée temporairement dans une zone mémoire de la machine (RAM). Elle peut être considérée comme une boîte de stockage. Elle est constituée de deux éléments :

- **son nom**, c'est l'étiquette de la boîte ;
- **sa valeur**, c'est le contenu de cette boîte. Ce contenu peut changer au cours de l'exécution des instructions.



Il existe des variables de **plusieurs types** (la liste n'est pas exhaustive).

- **Les chaînes de caractères** : leur contenu est du texte.

Exemple : le nom de la variable est *Jour* ; sa valeur initiale est : « lundi ».

- **Les nombres entiers** : leur contenu est du type 1, 2, 3, 4 ou encore 0, -1, -2...

- **Les nombres flottants** : leur contenu est un nombre décimal, par exemple 12,5.

Lorsque l'on donne une valeur à une variable X , on dit que l'on affecte une valeur à X , ce qui peut s'écrire :

Langage naturel

X prend la valeur ...

Pseudo-langage

$X \leftarrow \dots$

2 Qu'est-ce qu'une fonction ?

■ En mathématiques

Définir une fonction f de la variable x sur un ensemble D , c'est **associer à chaque valeur de la variable x** (c'est-à-dire chaque élément de D) **un et un seul nombre** noté $f(x)$.

En mathématiques, il existe des fonctions de plusieurs variables, mais elles ne sont pas abordées en classe de seconde.

Exemple : Pour tout x de l'intervalle $[-5 ; 5]$, $f(x) = (x - 1)^2 + 3$.

■ En algorithmique

Une fonction peut être considérée comme une séquence d'instructions réalisant une certaine tâche, en utilisant un ou plusieurs **arguments**. Cette fonction reçoit **un nom**.

Elle permet de découper le problème étudié en sous-problèmes et d'éviter ainsi la répétition inutile d'instructions.

Une fois une fonction définie, elle peut être « appelée » tout au long de l'exécution de l'algorithme, autant de fois que nécessaire.

Exemple : La fonction **compare** détermine le plus grand de deux nombres a et b donnés.

On peut ensuite « appeler » **compare()** dès que nécessaire dans la suite de l'algorithme, en identifiant bien les deux variables qui correspondent aux arguments (a et b).

Langage naturel

Fonction **compare** (a , b)
 Si $a \geq b$
 Alors retourne (a)
 Sinon retourne (b)
 FinSi

Étude qualitative de fonctions

Exercice 1 Variables et fonction

Variables et instructions élémentaires

On considère le script ci-contre écrit avec le logiciel Scratch.



1 Combien de variables (au sens algorithmique) ce script comporte-t-il ? Quels sont leurs noms ?

Ce script comporte deux variables algorithmiques, A et X.

2 Quel est le résultat affiché par le logiciel lorsque l'on saisit 2 pour valeur de X ?

$(2 \times 2)^2 + 2 \times 2 + 1 = 21$. Le résultat affiché est 21.

3 Quel est le résultat affiché par le logiciel lorsque l'on saisit (-1) pour valeur de X ?

$[2 \times (-1)]^2 + 2 \times (-1) + 1 = 3$. Le résultat affiché est 3.

4 Justifier que ce script définit une fonction f de la variable X. Donner une expression possible de f(X).

Ce script définit une fonction de la variable X, car à chaque valeur de X est associé un seul nombre noté f(X).

Une expression possible de f(X) est $4X^2 - 2X + 1$.

Exercice 2 Programmer une fonction mathématique

Variables et instructions élémentaires

On considère l'algorithme ci-contre écrit en langage naturel.

Langage naturel

Entrée	Saisir U
Traitement	Affecter à V la valeur $U^2 + 1$ Affecter à W la valeur 2V Affecter à X la valeur $W - 2$
Sortie	Afficher X

1 Combien de « variables » cet algorithme comporte-t-il ? De quel type de variable peut-il s'agir ?

Cet algorithme comporte 4 variables. Ce sont des flottants.

2 Exécuter cet algorithme lorsque la valeur de U est 3.

Préciser alors la valeur de chacune des variables à la fin de l'exécution de cet algorithme.

U = 3

V = 10

W = 20

X = 18

3 On donne ci-dessous la traduction de cet algorithme en trois langages de programmation. Implanter ce programme sur votre calculatrice ou sur un ordinateur. L'exécuter et vérifier que la valeur de X affichée correspond à la réponse donnée en 2.

T.I. Basic

```
PROGRAM:EXE
:Input "U=",U
:U^2+1→V
:2*V→W
:W-2→X
:Disp X
```

Basic Casio

```
"Entrer U"↵
?→U↵
U^2+1→V↵
2×V→W↵
W-2→X↵
X↵
```

Python

```
print("Entrer U:")
U=float(input())
V=U*U+1
W=2*V
X=W-2
print("X=",X)
```

4 Pour quelle(s) valeur(s) initiale(s) de U obtient-on $X = 98$? Justifier la réponse.

$X = 98 ; X = W - 2$ donc $W = X + 2 = 100 ;$

$W = 2V$ donc $V = \frac{W}{2} = 50 ;$

$U = \sqrt{V - 1} = 7.$

On obtient $X = 98$ pour $U = 7.$

Exercice 3 Calcul d'un temps de trajet

Instruction conditionnelle

Lorsque l'on fait une recherche d'itinéraire entre deux positions à l'aide d'un GPS, celui-ci nous renvoie une estimation de la durée du trajet. Cette durée est fonction, entre autres, du type de route empruntée et des limitations de vitesse sur ces routes. On rappelle ici que si l'on note D la distance en km d'un trajet, T le temps de trajet en h et V la vitesse moyenne en km/h, on a : $V = \frac{D}{T}$.



1 Compléter l'algorithme ci-contre pour qu'une fois la distance D et la vitesse moyenne V saisies, il calcule puis affiche le temps T de trajet correspondant.

Langage naturel

Variables	D, V, T nombres flottants..... (type de variable)
Entrées	Saisir D et V .
Traitement	Affecter à T la valeur D/V .
Sortie	Afficher T .

2 Quel est le résultat affiché par l'algorithme précédent si l'utilisateur se trompe et saisit : $D = -20$?

Si on saisit une distance négative, on va obtenir une durée négative.

3 On souhaite signaler à l'utilisateur, lorsque cela arrive, qu'il a fait une erreur de saisie. On utilise alors l'instruction conditionnelle « Si ...alors...sinon ». Compléter l'algorithme ci-contre de façon à ce qu'il réponde au problème de la question 1 mais signale l'erreur de saisie éventuelle.

Langage naturel

Variables	D, V, T nombres flottants.....
Entrées	Saisir D et V .
Traitement	Si $D \leq 0$ alors Afficher « La valeur de D doit être positive. Ressaisir les données. » Sinon Affecter à T la valeur D/V .
	FinSi
Sortie	Afficher T .

Exercice 4 Choisir le bon tarif

Boucle non bornée

Corentin est cinéophile. Il se demande s'il ne va pas opter pour la carte d'abonnement proposé par le cinéma le plus proche, seul cinéma dans lequel il se rend. Ce cinéma propose un Pass Mensuel qui coûte 20 € par mois et qui permet d'acheter les places à 9 € chacune. Sinon, le prix non réduit de la place est 12 €.

Il cherche à déterminer à partir de quel nombre de séances mensuelles ce Pass est intéressant.



1 Aidez-le à trouver sa réponse en complétant l'algorithme ci-contre.

Langage naturel

Variables N, S, P nombres entiers
Initialisation N prend la valeur 0
 S prend la valeur 0
 P prend la valeur 20
Traitement Tant que $S < P$
 N prend la valeur $N + 1$
 S prend la valeur $12N$
 P prend la valeur $20 + \dots 8 \dots N$
 FinTantQue
Sortie Afficher $\dots N \dots$

2 Sans utiliser de calculatrice ni d'ordinateur, compléter le tableau ci-dessous au fur et à mesure de l'exécution de l'algorithme précédent.

N	0	1	2	3	4	5	6	7
S	0	12	24	36	48	60	72	84
P	20	29	38	44	52	60	68	76
Condition $S < P$	Vérfiée	Vérfiée	Vérfiée	Vérfiée	Vérfiée	Non vérfiée	Non vérfiée	Non vérfiée

3 Quelle est la valeur affichée en sortie de cet algorithme ? Interpréter ce résultat pour répondre au problème de Corentin.

La valeur affichée en sortie de cet algorithme est 5. Il y a égalité des prix pour 5 séances par mois.

Le Pass est donc réellement intéressant pour Corentin à partir de la 6^e séance dans le même mois.

Exercice 5 Un peu d'économie

Boucle bornée

Le coût de production d'un produit, en milliers d'euros, est donné par la fonction mathématique $C(x) = \frac{1}{6}x^3 - \frac{5}{2}x^2 + 13x$, où x est la quantité fabriquée en tonnes. L'entreprise peut fabriquer entre 0 et 100 tonnes de ce produit. Chaque tonne est vendue 6 000 euros.

1 Compléter l'algorithme ci-contre afin qu'une fois la quantité x saisie, l'algorithme calcule et affiche le coût de production correspondant, en milliers d'euros.

Langage naturel

Variables X et C nombres flottants
Entrées Saisir $\dots X \dots$
Traitement Affecter à C la valeur $\frac{1}{6}X^3 - \frac{5}{2}X^2 + 13X$
Sortie Afficher $\dots C \dots$

2 Exprimer la recette $R(x)$ en milliers d'euros pour x tonnes vendues.

$R(x) = 6x$

3 Montrer que pour $x \in [0 ; 100]$, le bénéfice réalisé pour la vente de x tonnes est, en milliers d'euros :

$$B(x) = -\frac{1}{6}x^3 + \frac{5}{2}x^2 - 7x$$

$$B(x) = R(x) - C(x) = 6x - \frac{1}{6}x^3 + \frac{5}{2}x^2 - 13x = -\frac{1}{6}x^3 + \frac{5}{2}x^2 - 7x.$$

4 Quel est le rôle de l'algorithme suivant ?

Langage naturel

Variables X, B nombres flottants
Initialisation X prend la valeur 0
Traitement Pour i allant de 1 à 100
 X prend la valeur $X + 1$
 B prend la valeur $-\frac{1}{6}X^3 + \frac{5}{2}X^2 - 7X$
 Afficher B
 FinPour

Fin de l'algorithme

Pseudo-langage

$X \leftarrow 0$
 Pour i allant de 1 à 100
 $X \leftarrow X + 1$
 $B \leftarrow -\frac{1}{6}X^3 + \frac{5}{2}X^2 - 7X$
 Afficher B
 FinPour

Cet algorithme calcule les bénéfices pour toutes les valeurs de X entre 1 et 100.

5 On cherche à déterminer le bénéfice maximal réalisé par l'entreprise pour la vente de ce produit.

Compléter l'algorithme ci-contre afin qu'il réponde au problème.

Langage naturel

Variables X, B, M nombres flottants
Initialisation X prend la valeur 0
 M prend la valeur 0
Traitement Pour i allant de 1 à 100
 X prend la valeur $X + 1$
 B prend la valeur $-\frac{1}{6}X^3 + \frac{5}{2}X^2 - 7X$
 Si $B > M$
 M prend la valeur B
 FinSi
 FinPour
Sortie Afficher M

6 Écrire l'algorithme de la question 5 dans le langage de programmation de votre choix, et donner la valeur du bénéfice maximal (pour x entier) affichée par cet algorithme.

$B \approx 18,7$.



BESOIN D'AIDE?

Pour traduire l'algorithme, voir l'intérieur de la couverture.

T.I. Basic

$0 \rightarrow X$
 $0 \rightarrow M$
 For (1,1,100)
 $X+1 \rightarrow X$
 $-1/6 * X^3 + 5/2 * X^2 - 7 * X \rightarrow B$
 If $B > M$
 Then
 $B \rightarrow M$
 End
 End
 End
 Disp M

Basic Casio

$0 \rightarrow X$
 $0 \rightarrow M$
 For 1 \rightarrow I To 100
 $X+1 \rightarrow X$
 $-1 \div 6 \times X^3 + 5 \div 2 \times X^2 - 7 \times X \rightarrow B$
 If $B > M$
 Then $B \rightarrow M$
 IfEnd
 Next
 M

Python

$X=0$
 $M=0$
 for I in range (1,101):
 $X=X+1$
 $B=-1/6 * X**3 + 5/2 * X**2 - 7 * X$
 if $B > M$
 $M=B$
 print(M)

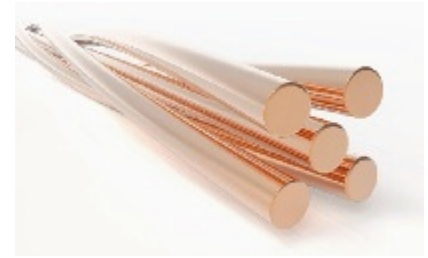


Résistance du cuivre

Notion de fonction

La résistance d'un fil de cuivre est fonction de la section S (en mm^2) et de la longueur L du fil (en m). On a $R = \frac{L}{57 \times S}$ (en ohms Ω).

Dans une salle de sciences physiques, 10 morceaux de fil de cuivre sont posés sur une paillasse. Pour chaque fil, les diamètres D des fils (en mm) ainsi que leurs longueurs L (en m) sont écrits sur une étiquette. L'idée est de concevoir un programme qui calcule et affiche la résistance, en Ω , de chacun de ces fils.



$D = 0,5 \text{ mm}$ $L = 0,02 \text{ m}$	$D = 0,6 \text{ mm}$ $L = 0,03 \text{ m}$	$D = 0,8 \text{ mm}$ $L = 0,05 \text{ m}$	$D = 1,0 \text{ mm}$ $L = 0,05 \text{ m}$	$D = 1,0 \text{ mm}$ $L = 0,08 \text{ m}$	$D = 1,2 \text{ mm}$ $L = 0,08 \text{ m}$	$D = 1,3 \text{ mm}$ $L = 0,05 \text{ m}$	$D = 1,4 \text{ mm}$ $L = 0,05 \text{ m}$	$D = 1,3 \text{ mm}$ $L = 0,06 \text{ m}$	$D = 1,5 \text{ mm}$ $L = 0,09 \text{ m}$
Fil n° 1	Fil n° 2	Fil n° 3	Fil n° 4	Fil n° 5	Fil n° 6	Fil n° 7	Fil n° 8	Fil n° 9	Fil n° 10

PARTIE 1

Un script qui calcule et détermine la résistance en fonction de L et D

1 Donner l'expression de la section S (en mm^2) d'un fil en fonction de son diamètre D (en mm).

$$S = \pi \times (D/2)^2$$

2 À chaque couple (L, D) correspond une valeur de résistance R .

Compléter le script ci-contre en langage Python et implanter ce programme sur un ordinateur.

```
D=float(input("diamètre D="))
L=float(input("longueur L="))
S=3.14159*((D/2)**2)
R=L/(57*S)
print("La résistance en ohms est environ")
print (R)
```

3 En utilisant les données inscrites sur les 10 étiquettes posées sur la paillasse dans la salle et votre programme précédent, déterminer le fil dont la résistance est la plus grande.

Le fil dont la résistance est la plus grande est le fil numéro 2.

PARTIE 2

Une fonction à deux arguments L et D que l'on appelle si nécessaire

On souhaite à présent définir une fonction (au sens algorithmique) de deux arguments : L la longueur du fil (en m) et D son diamètre (en mm).

L'objectif est de concevoir un programme plus efficace et éviter ainsi d'exécuter 10 fois le programme précédent.

4 Définir la fonction `resist` en complétant le script ci-contre en langage Python.

```
def resist(L,D):
    S=3.14159*((D/2)**2)
    R=L/(57*S)
    return R
```

5 Le script complet d'un programme à l'aide duquel il est possible d'obtenir les résistances des 10 fils posés sur la paillasse est en partie donné ci-contre. Compléter ce script.

```
def resist(L,D):
    S=3.14159*((D/2)**2)
    R=L/(57*S)
    return R

for I in range(1,11):
    print ("Fil numéro")
    print (I)
    D=float(input("Diamètre D="))
    L=float(input("Longueur L="))
    resist(L,D)
    print ("a pour résistance")
    print (resist(L,D))
```

Une fois le script vérifié par le professeur, l'implanter sur l'ordinateur et utiliser ce programme pour afficher les résistances des 10 fils disponibles et compléter le tableau ci-dessous.

Fil n° 1	Fil n° 2	Fil n° 3	Fil n° 4	Fil n° 5
0,00179	0,00186	0,00174	0,00112	0,00179
Fil n° 6	Fil n° 7	Fil n° 8	Fil n° 9	Fil n° 10
0,00124	0,00066	0,00056	0,00079	0,00089

PARTIE 3 Trouver le numéro du fil dont la résistance est maximale

6 Modifier le script suivant pour qu'il affiche, une fois les dimensions des 10 fils saisis, le numéro N du fil dont la résistance est la plus grande, ainsi que la valeur M de cette résistance maximale (en Ω).

```
def resist(L,D):
    S=3.14159*((D/2)**2)
    R=L/(57*S)
    return R

M=0
for I in range(1,11):
    print ("Fil numéro")
    print (I)
    D=float(input("Diamètre D="))
    L=float(input("Longueur L="))
    resist(L,D)
    print("a pour résistance")
    print(resist(L,D))
    if resist(L,D)>M:
        M=resist(L,D)
        N=I
print ("le numéro du fil dont la résistance est maximale est")
print (N)
print ("Sa résistance est de :")
print (M)
print ("ohms")
```

7 Implanter ce programme sur un ordinateur, puis donner le numéro du fil N qui a la résistance maximale, ainsi que la valeur en ohms de cette résistance maximale.

$N = 2$ et $R \approx 0,001861 \Omega$ (arrondir cette valeur à 10^{-6}).

Expressions algébriques

Exercice 1 **Égal ou non ?** Variables et instructions élémentaires

Anna et Younès ont écrit les deux scripts suivants avec le logiciel Scratch.

Script Scratch d'Anna

```

quand espace est pressé
  demander Entrez une valeur de X et attendez
  mettre X à réponse
  mettre A à X + 1
  mettre B à X - 3
  mettre Y à A * B
  dire Y
          
```

Script Scratch de Younès

```

quand espace est pressé
  demander Entrez une valeur de X et attendez
  mettre X à réponse
  mettre U à X * X
  mettre V à -2 * X
  mettre W à U + V + -3
  dire W
          
```

1 a) Sans utiliser le logiciel Scratch, déterminer la valeur de Y affichée par le script d'Anna pour $X = 2$.

$$Y = (2 + 1) \times (2 - 3) = -3.$$

b) De la même manière, déterminer la valeur de W affichée par le script de Younès pour $X = 2$.

$$W = 2^2 - 2 \times 2 - 3 = -3.$$

c) Que remarque-t-on ?

Les valeurs de Y et de W sont égales pour $X = 2$.

2 Sarah affirme que pour toute valeur de X saisie, les valeurs des variables Y et W affichées par ces deux scripts sont égales.

Sarah a-t-elle raison ? Justifier la réponse.

Le script d'Anna peut s'écrire $(X + 1)(X - 3)$. Si on développe cette expression on obtient $X^2 - 3X + X - 3 = X^2 - 2X - 3$.

Le script de Younès s'écrit $X^2 - 2X - 3$. Les deux scripts donneront donc le même résultat quelle que soit la valeur de X saisie.

Sarah a raison.

3 Sarah affirme également que chacun de ces scripts peut s'écrire à l'aide de deux variables seulement. En ne créant que deux variables X et Y , proposer un script qui renvoie pour toute valeur de X la même valeur de Y que le script d'Anna.

Scratch

Quand espace est pressé

Demander « Entrez une valeur de X » et attendez

Mettre X à réponse

Mettre Y à $X^2 - 2X - 3$

Dire Y

Exercice 2

Retrouver un antécédent

Variables et instructions élémentaires

On donne ci-dessous deux copies d'écran de calculatrices et une copie d'un script en langage Python. Ces trois programmes correspondent à un seul algorithme.

T.I. Basic

```
PROGRAM: CALCUL
: Input "X=", X
: X-3→A
: A^2→A
: A-X^2→A
: Disp A
```

Basic Casio

```
===== CALCUL =====
"Saisir X"↵
?→X↵
X-3→A↵
A^2→A↵
A-X^2→A↵
A↵
```

Python

```
X=float(input("X="))
A=X-3
A=A*A
A=A-X*X
print(A)
```

1 Combien ces programmes comportent-ils de variables (au sens algorithmique) ?

Ils comportent 2 variables au sens algorithmique.

2 Déterminer le résultat affiché par la calculatrice ou l'ordinateur lorsque $X = -1$.

$(-1 - 3)^2 - (-1)^2 = 15$.

3 Cet algorithme définit une fonction de la variable réelle X . Donner l'expression de $f(X)$ en fonction de X .

$f(X) = (X - 3)^2 - X^2 = X^2 - 6X + 9 - X^2 = -6X + 9$.

4 Après exécution de ce programme, on obtient à l'affichage $A = 10$. Quelle était la valeur de X saisie au départ ?

$-6X + 9 = 10$;

$-6X = 1$;

$X = -\frac{1}{6}$.

5 Écrire un algorithme en langage naturel permettant de retrouver l'antécédent d'un nombre A par la fonction f .

Langage naturel

Variables : A, X sont des nombres flottants.

Entrée : Saisir A .

Traitement : Affecter à X la valeur $-\frac{A-9}{6}$.

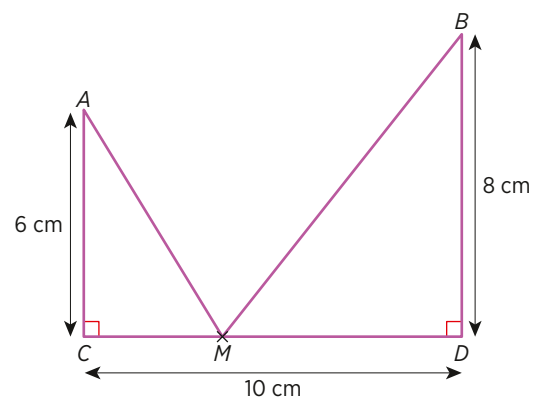
Sortie : Afficher X .

Exercice 3

Déterminer des distances

Instruction conditionnelle et boucle bornée

Dans la configuration ci-contre, M est un point mobile du segment $[CD]$, et les triangles ADM et BCM sont rectangles. On cherche à déterminer s'il existe un point M tel que la distance CM soit un nombre entier et $MA = MB$. On pose $x = DM$.



1 À quel intervalle appartient la variable x ? Donner l'intervalle d'amplitude maximale.

La variable x appartient à l'intervalle $[0 ; 10]$.

- 2 On notera a la borne inférieure de cet intervalle et b sa borne supérieure. Compléter l'algorithme suivant pour qu'il détermine si un nombre X saisi appartient à cet intervalle $[a ; b]$.

Langage naturel

Variables	X nombre flottant R de type chaîne de caractères
Entrée	Saisir X
Traitement	Si $X \geq a$ <u>et</u> $X \leq b$ Affecter à R la valeur « Le nombre saisi <u>appartient à l'intervalle $[a ; b]$</u> » Sinon affecter à R la valeur « Le nombre saisi <u>n'appartient pas à l'intervalle $[a ; b]$</u> » FinSi
Sortie	Afficher R

- 3 Compléter l'algorithme ci-contre afin qu'il calcule, pour chaque valeur de X de l'intervalle $[a ; b]$ déterminé à la question précédente, les distances AM et BM , puis qu'il compare ces distances et précise si elles sont égales ou non.

Langage naturel

Variables	X nombre flottant U nombre flottant V nombre flottant R chaîne de caractères
Entrée	Saisir X
Traitement	Si $X \geq a$ <u>et</u> $X \leq b$ Affecter à U la valeur $\sqrt{X^2 + 36}$ Affecter à V la valeur $\sqrt{(10 - X)^2 + 64}$ Si <u>$U = V$</u> Affecter à R la valeur « M est équidistant de A et B » Sinon Affecter à R la valeur « M n'est pas équidistant de A et B » FinSi Sinon Affecter à R la valeur « Valeur de X impossible » FinSi
Sortie	Afficher R

- 4 a) En utilisant l'algorithme précédent, écrire un algorithme en langage naturel qui, pour chaque position du point M tel que CM soit un nombre entier, détermine si ce point M est équidistant des points A et B .

Langage naturel

Variables : U nombre flottant

V nombre flottant

R chaîne de caractères

Traitement : Pour I allant de 0 à 10

Affecter à U la valeur $\sqrt{I^2 + 36}$

Affecter à V la valeur $\sqrt{(10 - I)^2 + 64}$

Si $U = V$

Affecter à R la valeur « CM est un entier et M est équidistant de A et B »

Sinon

Affecter à R la valeur « Il n'y a pas de solution »

FinSi

Sortie : Afficher R



BESOIN D'AIDE?

On utilisera une boucle bornée « Pour I allant de 0 à 10. ».

b) Compléter à l'aide de cet algorithme le tableau suivant.

Distance <i>CM</i> (cm)	0	1	2	3	4	5	6	7	8	9	10
Distance <i>AM</i> (cm)	6	6,08	6,32	6,71	7,21	7,81	8,48	9,22	10	10,82	11,66
Distance <i>BM</i> (cm)	12,8	12,04	11,31	10,63	10	9,43	8,94	8,54	8,25	8,06	8

c) Existe-t-il un (ou plusieurs) entier(s) qui réponde(nt) au problème ?

Non, il n'existe pas d'entiers qui répondent au problème.

Exercice 4 Un engagement pour la planète Boucle bornée

Un grand groupe industriel a rejeté 6 000 tonnes de déchets dans l'atmosphère sur l'année 2016. Le PDG a signé fin 2016 un contrat l'engageant à réduire chaque année les déchets rejetés par l'entreprise de 5 % par an dès 2017. À terme, il doit promettre de limiter ces déchets annuels à 1 500 tonnes.

1 Si l'on note D la quantité de déchets, en tonnes, rejetés en une année, exprimer en fonction de D la quantité maximale (en tonnes) rejetée l'année suivante.

La quantité de déchets rejetés l'année suivante étant réduite de 5 % chaque année, elle s'écrit $D - 0,05 \times D = 0,95 \cdot D$.

2 On donne l'algorithme ci-contre.

a) À quoi correspond la variable N ? L'année.

b) À quoi correspond la variable Q ?
La quantité de déchets rejetés (en tonnes) l'année N .

c) Quelle est la valeur de N à la fin de l'exécution de cet algorithme ?

À la fin de l'algorithme, N vaut $2016 + 10 = 2026$.

d) À quoi correspond la valeur de Q à la fin de l'exécution de cet algorithme ?

Elle correspond à la quantité de déchets rejetés en 2026.

Pseudo-langage

```

Q ← 6 000
N ← 2016
Pour i allant de 1 à 10
  N ← N + 1
  Q ← 0,95 × Q
FinPour
    
```

3 Implanter l'un des programmes suivant selon le modèle de votre calculatrice.

Exécuter ce programme et donner une interprétation du résultat pour le PDG de ce groupe.

T.I. Basic

```

PROGRAM:ECOLO
:6000→Q
:2016→N
:While Q≥1500
:N+1→N
:0.95*Q→Q
:End
:Disp N
    
```

Basic Casio

```

===== ECOLO =====
6000→Q↵
2016→N↵
While Q≥1500↵
N+1→N↵
0.95×Q→Q↵
WhileEnd↵
N↵
↵
    
```

Ce programme donne l'année où la quantité de déchets rejetés atteindra 1 500 tonnes, c'est-à-dire l'année 2044.

Le PDG du groupe aura alors atteint son objectif.

Exercice 5

Les logiciels de calcul formel sont-ils « intelligents » ?

Variables et instructions élémentaires

Un logiciel de calcul formel (ici Xcas) facilite le calcul symbolique et, par exemple, peut développer ou factoriser une expression algébrique donnée (voir copie ci-dessous) :

Ligne 1 `developper ((2*x-1)*(3*x+2))`
 Ligne 2 `6*x2+x-2`

Le but de cet exercice est d'automatiser un développement d'expression de type $f(x) = (ax + b)(cx + d)$ où a, b, c et d sont des nombres réels constants.



1 Pour tout réel x , montrer que $f(x) = acx^2 + (ad + bc)x + bd$.

$$(ax + b)(cx + d) = acx^2 + adx + bcx + bd = acx^2 + (ad + bc)x + bd.$$

2 Quel est le rôle de l'algorithme ci-dessous ?

Langage naturel

Variables	a, b, c, d, e, f et g nombres flottants
Entrées	Saisir a, b, c et d
Traitement	Affecter à e la valeur $(a \times c)$ Affecter à f la valeur $(a \times d + b \times c)$ Affecter à g la valeur $(b \times d)$
Sortie	Afficher e, f et g

Cet algorithme calcule les coefficients du développement de l'expression $(ax + b)(cx + d)$.

3 En traduisant un algorithme construit à partir du précédent en langage Python, on obtient le script suivant.

Python

```
a=float(input("Entrer la valeur de a="))
b=float(input("Entrer la valeur de b="))
c=float(input("Entrer la valeur de c="))
d=float(input("Entrer la valeur de d="))
e=a*c
f=a*d+b*c
g=b*d
print ("Le développement de l'expression (ax+b)(cx+d) est:")
print (e, "x*x+", f, "x+", g)
```

Implanter ce programme sur un ordinateur et vérifier que l'on retrouve le résultat donné en Ligne 2 de l'instruction saisie en Ligne 1 sur la copie d'écran du logiciel de calcul formel.

4 Compléter le script suivant afin qu'il automatise le développement d'expressions du type $(ax + b)^2$.

Python

```
a=float(input("Entrer la valeur de a="))
b=float(input("Entrer la valeur de b="))
c= a**2.....
d= 2*a*b.....
e= b**2.....
print("Le développement de l'expression (ax+b)**2 est:")
print(c, "x*x+", d, "x+", e)
```



Comparaisons d'expressions algébriques

Notion de fonction

On considère les trois fonctions d'arguments a et b définies ci-dessous en langage Python.

```
def fonction1(a,b):
    return((a+b)**2)
```

```
def fonction2(a,b):
    return((a-b)**2)
```

```
def fonction3(a,b):
    return(2*(a**2+b**2))
```



BESOIN D'AIDE?

En langage Python, « x puissance y » se note « x**y ».

PARTIE 1 Calculs

Répondre aux questions suivantes sans implanter de script sur l'ordinateur.

- Quelle est la valeur retournée par la fonction 1 pour $a = -2$ et $b = 5$? $(-2 + 5)^2 = 9$.
- Quelle est la valeur retournée par la fonction 2 pour $a = -2$ et $b = 5$? $(-2 - 5)^2 = 49$.
- Quelle est la valeur retournée par la fonction 3 pour $a = -2$ et $b = 5$? $(2 \times ((-2)^2 + 5^2)) = 2 \times (4 + 25) = 58$.

PARTIE 2 Comparaison d'expressions algébriques

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

- Implanter et exécuter ce programme pour :
 - $a = -2$ et $b = 5$: on a $z = -40$.
 - $a = 4$ et $b = -6$: on a $z = -96$.
 - $a = 6$ et $b = 5$: on a $z = 120$.
- Prouver que pour tous entiers positifs a et b , la valeur affichée de z à la sortie est un multiple de 4.

$$z = x - y = (a + b)^2 - (a - b)^2 = a^2 + 2ab + b^2 - a^2 + 2ab - b^2 = 4ab.$$

z est donc un multiple de 4.

```
def fonction1(a,b):
    return((a+b)**2)

def fonction2(a,b):
    return((a-b)**2)

a=float(input("a="))
b=float(input("b="))

x=fonction1(a,b)
y=fonction2(a,b)
z=x-y

print(z)
```

PARTIE 3 Test d'égalité d'expressions algébriques

- On considère le nouveau script ci-contre. L'exécuter sur un ordinateur puis compléter le tableau suivant au fur et à mesure de l'exécution.

Valeur de l	1	2	3	4
Valeur de a à saisir	2	-3	2	65
Valeur de b à saisir	-5	7	10	23
Valeur de x	9	16	144	7 744
Valeur de y	49	100	64	1 764
Valeur de z	58	116	208	9 508
Condition (V ou F)	V	V	V	V
Affichage (« oui ou non »)	oui	oui	oui	oui

```
def fonction1(a,b):
    return((a+b)**2)

def fonction2(a,b):
    return((a-b)**2)

def fonction3(a,b):
    return(2*(a**2+b**2))

for l in range(1,5):
    a=float(input("a="))
    b=float(input("b="))
    x=fonction1(a,b)
    y=fonction2(a,b)
    z=fonction3(a,b)
    if x+y==z:
        print("oui")
    else:
        print("non")
```

- La condition $x+y=z$ est-elle toujours vérifiée ? Justifier soigneusement la réponse.

La condition est toujours vérifiée car $(a + b)^2 + (a - b)^2 = a^2 + 2ab + b^2 + a^2 - 2ab + b^2 = 2(a^2 + b^2) = z$.

Équations - inéquations

Exercice 1 Résolution d'une équation

Instruction conditionnelle

On considère le script ci-contre écrit avec le logiciel Scratch.

1 Combien de variables ce script comporte-t-il ? Quels sont leurs noms ?


Ce script contient la variable a et la variable s .

2 Écrire et exécuter ce script, en donnant 25 comme valeur de a . Que renvoie le programme ?

Le programme renvoie la phrase suivante : « les deux solutions sont 5.0 et -5.0 ».

3 Quelle est l'équation que ce script cherche à résoudre ?

La présence du bloc « mettre s à racine de a », ainsi que le test effectué à la question 2., indiquent que le programme cherche à résoudre l'équation $x^2 = a$.

4 Le bloc  permet d'obtenir l'opposé du nombre s . Quel autre bloc aurait-on pu utiliser ?

On aurait pu utiliser le bloc $[-1*s]$.

Scratch

```

quand est cliqué
demander "Entrez la valeur de a : " et attendre
mettre a à réponse
si a < 0 alors
  dire "il n'y a pas de solution" pendant 2 secondes
si a = 0 alors
  dire "0 est la seule solution" pendant 2 secondes
si a > 0 alors
  mettre s à racine de a
  dire "les deux solutions sont : " pendant 1 secondes
  dire s pendant 1 secondes
  dire "et " pendant 1 secondes
  dire [-s] pendant 1 secondes
  
```

Exercice 2 Solutions d'une inéquation

Instruction conditionnelle

Simon souhaite acheter, avec les 3 € dont il dispose, des sucettes (à 0,35 € l'unité) ainsi que des chewing-gums (à 0,20 € l'unité).

1 En appelant s le nombre de sucettes et c le nombre de chewing-gums, écrire l'inéquation que s et c doivent vérifier pour que Simon puisse acheter s sucettes et c chewing-gums.

Il faut que s et c vérifient : $0,35s + 0,2c \leq 3$.

2 Compléter le programme en langage naturel ci-dessous afin qu'il dise à Simon si son choix est possible ou non.

Langage naturel

Variables	S est un nombre entier C est un nombre entier
Entrées	Saisir S , C
Traitement	Si $0,35*S + 0,2*C$ est inférieur ou égal à 3 : Afficher « choix possible » Sinon Afficher « choix impossible »



Exercice 3

Recherche par force brute d'une solution entière

Boucle non bornée

On considère l'équation $x^3 - 3x^2 + x = 3\,344$. On suppose que cette équation a une solution positive entière, qu'il faut chercher.

1 Tester cette équation pour $x = 10$ puis pour $x = 20$.

$10^3 - 3 \times 10^2 + 10 = 710$ et $20^3 - 3 \times 20^2 + 20 = 6\,820$ donc la solution cherchée n'est ni 10 ni 20.

2 Pour essayer de trouver une solution à cette équation, on décide de tester tous les nombres entiers positifs, jusqu'à ce qu'on trouve (peut-être) la solution. On appelle cette méthode la méthode de « force brute ». On considère donc le programme en langage naturel ci-contre.

Langage naturel

Variables	X est un nombre entier
Entrées	Affecter à X la valeur 0
Traitement	Tant que $X^3 - 3X^2 + X$ est différent de 3 344 : Affecter à X la valeur $X + 1$
	FinTantQue
Sortie	Afficher X

Que se passera-t-il si l'équation $x^3 - 3x^2 + x = 3\,344$ n'a pas de solution entière ?

Le programme ne sortira jamais de la boucle conditionnelle « Tant que ».

3 Le programme ci-dessus a été traduit en plusieurs langages.

Python

```
X=0
while X**3-3*X**2+X!=3344:
    X=X+1
print(X)
```

T.I. Basic

```
PROGRAM:PROG
:0→X
:While X^3-3X^2+
X≠3344
:X+1→X
:End
:DISP X
```

Basic Casio

```
=====PROG=====
0→X
While X^3-3X^2+X≠3344
  X+1→X
WhileEnd
X
```

Implanter cet algorithme dans le langage de votre choix. Quelle est la solution de l'équation $x^3 - 3x^2 + x = 3\,344$?

La solution est $x = 16$.

Exercice 4

Tableau de signes automatique d'une fonction affine

Instruction conditionnelle

Cet exercice a pour but de construire automatiquement le tableau de signes de n'importe quelle fonction affine $f(x) = ax + b$, afin d'aider à la résolution d'inéquations du type $ax + b > 0$.

1 Construire le tableau de signes de $f(x) = -5x + 2$.

En déduire l'ensemble de solutions de l'inéquation $-5x + 2 \geq 0$.

Il faut d'abord calculer la valeur charnière $c = \frac{-b}{a}$. Ici, $a = -5$ et $b = 2$. Donc la valeur charnière c est égale à $\frac{-2}{(-5)} = \frac{2}{5} = 0,4$.

De plus, $a < 0$ donc le tableau de signes est :

x	$-\infty$	$0,4$	$+\infty$
$f(x)$	$+$	0	$-$

Par lecture de ce tableau de signes, l'inéquation $-5x + 2 \geq 0$ a pour solution $S =]-\infty ; 0,4]$.

.... RAPPEL

Soit $f(x) = ax + b$ une fonction affine (avec $a \neq 0$).

• si $a < 0$, le tableau de signes de f est

x	$-\infty$	$-\frac{b}{a}$	$+\infty$
$f(x)$	$+$	0	$-$

• si $a > 0$, le tableau de signes de f est

x	$-\infty$	$-\frac{b}{a}$	$+\infty$
$f(x)$	$-$	0	$+$

2 Compléter le programme Python ci-contre pour qu'il construise automatiquement le tableau de signes de $f(x) = ax + b$, en fonctions des variables a et b rentrées par l'utilisateur.

3 Utiliser votre programme pour résoudre l'inéquation $-14x - 98 < 0$.

D'après notre programme, $f(x)$ est positif sur $]-\infty, -7]$ et strictement négatif sur l'intervalle $]-7; +\infty[$; l'inéquation $-14x - 98 < 0$ a donc pour solution l'intervalle $]-7; +\infty[$.

Python

```
a=float(input("a=?"))
b=float(input("b=?"))

#calcul de la valeur charnière
c=-b/a # <- à compléter

#construction de la première ligne du tableau
print("-----")
print("|x   |      ",c,"   |")
print("-----")

#construction de la deuxième ligne du tableau
if a<0 : # <- à compléter
    print("|f(x)|   +   0   -   | ")
else :
    print("|f(x)|   -   0   +   | ")

print("-----")
```

Exercice 5 La rencontre des comètes

Boucle bornées

La comète Tchouri est passée dans notre système solaire pour la dernière fois en 2015. Elle a une période de révolution de 7 ans, ce qui signifie qu'elle repassera dans notre système solaire en 2022, 2029, etc. La comète Encke qui est passée pour la dernière fois en 2017, a elle une période de révolution de 3 ans. On cherche à déterminer en quelle année, au plus tôt, les deux comètes Tchouri et Encke traverseront simultanément notre système solaire.



La comète Tchouri.

1 On appelle a le nombre de révolutions de la comète Tchouri, et b le nombre de révolutions de la comète Encke. Écrire l'équation que doivent vérifier a et b pour que les deux comètes traversent le système solaire la même année.

Après a révolutions, Tchouri traverse le système solaire à l'année $2015 + 7 \times a$.

Après b révolutions, Encke traverse le système solaire à l'année $2017 + 3 \times b$.

Pour que ces deux comètes traversent le système solaire la même année, il faut donc que a et b vérifient l'équation

$2015 + 7 \times a = 2017 + 3 \times b$.

2 Pour résoudre cette équation, on propose le programme Python ci-contre. L'algorithme utilisé consiste à tester l'équation du 1. avec toutes les valeurs possibles a et b entre 0 et 9. Compléter ce programme.

Python

```
for a in range(10):
    for b in range(10):
        if 2015+7*a==2017+3*b:
            print(a,b)
```

3 Exécuter ce programme et en déduire la prochaine année de rencontre de ces deux comètes dans le système solaire.

Le programme ci-dessus renvoie les valeurs : 2 4. Cela signifie que $a = 2$ et $b = 4$. La rencontre aura donc lieu quand Tchouri aura fait 2 révolutions (depuis 2015). Ceci correspond à l'année $2015 + 7 \times 2$, soit en 2029. On peut vérifier que $2017 + 4 \times 3 = 2029$, donc Encke sera bien présente dans le système solaire, après 4 révolutions.



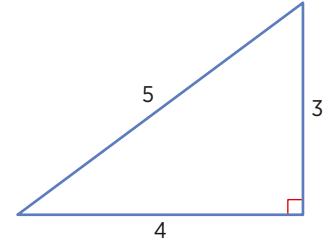
À la recherche des triplets pythagoriciens

Boucle bornée, optimisation

Le théorème de Pythagore nous assure que le triangle dont les côtés mesurent 3 cm, 4 cm et 5 cm est un triangle rectangle, car $3^2 + 4^2 = 5^2$. Les trois nombres (3, 4, 5) forment un triplet de nombres entiers appelé triplet pythagoricien. (8, 15, 17) est un autre triplet pythagoricien, car $8^2 + 15^2 = 17^2$.

Plus généralement, un triplet (a, b, c) sera dit pythagoricien si les trois nombres entiers a, b et c vérifient l'équation $a^2 + b^2 = c^2$. On classe généralement les nombres a, b et c dans l'ordre croissant.

On souhaite construire un programme permettant de trouver tous les triplets pythagoriciens inférieurs à une borne maximale.



PARTIE 1 Première approche

Dans cette partie, on cherche tous les triplets pythagoriciens constitués de nombres strictement inférieurs à 20. L'algorithme le plus évident consiste à tester toutes les valeurs a, b et c comprises entre 1 et 20, et à tester à chaque fois si $a^2 + b^2 = c^2$.

1 Traduire en langage Python l'algorithme ci-dessous :

Pseudo-langage

```
nbmax ← 20
pour a allant de 1 à nbmax :
  pour b allant de 1 à nbmax :
    pour c allant de 1 à nbmax :
      si  $a^2 + b^2 = c^2$  :
        afficher (a, b, c)
```



BESOIN D'AIDE?

En Python, pour qu'une variable i parcoure tous les nombres entiers compris entre 13 (inclus) et 20 (exclu), on utilise le code suivant :
for i in range(13,20):

Python

```
nbmax=20
for a in range(1,nbmax):
  for b in range(1,nbmax):
    for c in range(1,nbmax):
      if a**2+b**2==c**2:
        print(a,b,c)
```

2 Exécuter votre code Python et observer les résultats. Quelle remarque peut-on faire ?

Les triplets obtenus sont : (3, 4, 5), (4, 3, 5), (5, 12, 13), (6, 8, 10), (8, 6, 10), (8, 15, 17), (9, 12, 15), (12, 5, 13), (12, 9, 15), (15, 8, 17)...

Nous pouvons remarquer que cette liste comporte des triplets identiques : par exemple, les deux premiers triplets (3, 4, 5) et (4, 3, 5) sont les mêmes. On dit que ce sont des doublons.

PARTIE 2 Élimination des doublons

Pour éliminer les doublons, il faudrait faire en sorte que les trois nombres a, b et c qui composent le triplet soient rangés dans l'ordre croissant : $a < b < c$.

3 Rajouter une ligne à ce code pour que les triplets s'affichent uniquement si $a < b < c$.

On rajoute la ligne « if a<b<c: » entre les lignes « if a**2+b**2==c**2: » et « print(a,b,c) ».

Les doublons sont bien éliminés, chaque triplet est maintenant unique.

- 4 Mettre la valeur de nbmax à 1 000 et exécuter le code. Quel commentaire peut-on faire ?

L'exécution du code est très lente.



BESOIN D'AIDE?

Pour interrompre l'exécution d'un code, taper CTRL-C.

PARTIE 3 Optimisations de l'algorithme

La lenteur du code est due aux trois boucles *for* imbriquées. Dans cette partie, la valeur de nbmax est fixée à 1 000. Chaque variable a , b ou c prend donc les valeurs de 1 à 999.

- 5 Combien de triplets (a, b, c) sont testés par le code ?

$999 \times 999 \times 999 = 997\,002\,999$. Le nombre de triplets testés est environ égal à 1 milliard.

- 6 Pour éviter les doublons, on ne garde que les triplets vérifiant $a < b < c$. Dans la Partie 2, tous les triplets non valides ont été écartés par le test « if $a < b < c$: ». Or il est possible de ne garder que les triplets vérifiant $a < b < c$ par une simple modification des trois boucles initiales. Le programme devient plus rapide. Compléter le programme ci-contre.

Python

```
nbmax=1000
for a in range(1,nbmax):
    for b in range(.a., nbmax):
        for c in range(.b., nbmax):
            if a**2+b**2==c**2:
                print(a,b,c)
```

- 7 Le code peut être encore optimisé en supprimant la boucle concernant la variable c . En effet, une fois a et b choisis, le triplet pythagorien existera si et seulement si $\sqrt{a^2 + b^2}$ est un nombre entier. Traduire en langage Python l'algorithme ci-dessous.

Pseudo-langage

```
nbmax ← 1 000
pour a allant de 1 à nbmax :
    pour b allant de a à nbmax :
        c ←  $\sqrt{a^2 + b^2}$ 
        si c est un nombre entier :
            affiche (a, b, c)
```



BESOIN D'AIDE?

- Pour utiliser la fonction racine carrée `sqrt()`, il faut écrire en début de programme `from math import sqrt`.
- Pour tester si un nombre c est entier, on peut tester si il est égal à sa partie entière `int(c)`.

Python

```
from math import sqrt
nbmax=1000
for a in range(1,nbmax):
    for b in range(a,nbmax):
        c=sqrt(a**2+b**2)
        if c==int(c):
            print(a,b,c)
```

- 8 Comparer la vitesse d'exécution de ce code avec celui obtenu à la fin de la partie 2.

Ce code est beaucoup plus rapide. Le gain essentiel a été obtenu en supprimant la boucle concernant la variable c .

- 9 On souhaite construire un triangle rectangle dont les côtés sont des nombres entiers, et dont le plus petit côté mesure 893 cm. Combien mesurent les deux autres côtés ?

Parmi les derniers triplets pythagoriens donnés par notre programme, on trouve le triplet (893, 924, 1 285).

Les deux autres côtés mesurent donc 924 cm et 1 285 cm.

Fonctions de référence : fonctions affines et linéaires, fonction carré, fonction inverse

Exercice 1

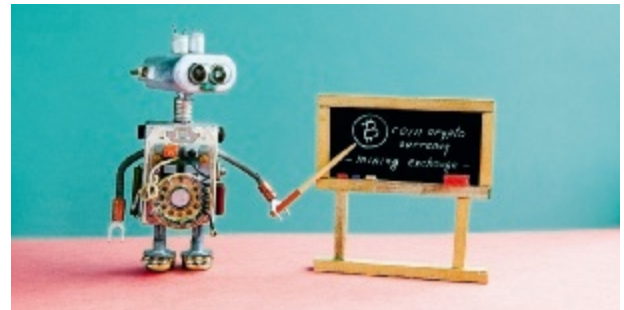
Convertisseur bitcoins-euros

Instruction conditionnelle

« Qu'est-ce que le bitcoin ?

Le bitcoin est la toute première monnaie virtuelle, créée en 2009 par un inventeur inconnu. Si l'on ne sait pas vraiment qui est à l'origine de cette monnaie, on connaît l'origine de son nom. Celui-ci vient de bit – une unité de mesure en informatique – et de coin, qui signifie monnaie en anglais. »

Le Monde.fr – 11 octobre 2017



Le site *bitcoin.fr* permet de connaître en temps réel le cours de cette monnaie virtuelle. Au 28 octobre 2017, on pouvait lire sur l'une des pages du site :

Prix du bitcoin (moyenne sur 24 h)
EUR 4 984 – USD 5 786 – CAD 7 413

1 Donner la somme en euros correspondant à 22 bitcoins à cette date.

$$22 \times 4\,984 = 109\,648 \text{ €}$$

2 On souhaite élaborer un programme qui convertit une somme donnée en bitcoins (notée B) en sa valeur en euros (notée E). Compléter l'algorithme ci-contre pour qu'il calcule et affiche, une fois la valeur de B saisie, la valeur de E correspondante.

Langage naturel

Variables	E, B nombres flottants
Entrée	Saisir B
Traitement	Affecter à E la valeur $4\,984 \times B$
Sortie	Afficher E

3 Traduire l'algorithme ci-dessus en un script dans le logiciel Scratch, et l'implanter sur un ordinateur. Vérifier qu'il fonctionne en retrouvant la réponse à la question 1.

Scratch

Quand espace est pressé
Demander B et attendre
Mettre B à réponse
Mettre E à $4984 \times B$
Dire E

4 Exécuter le programme précédent pour compléter le tableau ci-dessous.

Somme en bitcoins B	30	35	40
Somme en euros E	149 520	174 440	199 360

5 La somme E en euros est fonction de la somme B en bitcoins correspondante.

a) Exprimer E en fonction de B .

$$E = 4\,984 \times B$$

b) De quel type de fonction s'agit-il ?

Il s'agit d'une fonction linéaire.

c) Quelle est la valeur en euros d'une somme de 100 000 bitcoins ?

100 000 bitcoins valent 498 400 000 euros.

6 De nombreux logiciels ou sites permettent d'effectuer les conversions entre différentes « devises ». On souhaite ici écrire un programme qui effectue la conversion de bitcoins en euros ou d'euros en bitcoins. On commence pour cela par demander à l'utilisateur la devise dans laquelle il va saisir la somme au départ. Il répond « bitcoin » ou « euro ». On considère le script suivant.

a) Combien de variables comporte ce script ? Préciser le nom et le type de chaque variable.

Ce script comporte deux variables, E et B , qui sont des flottants.

b) Ce script est-il correct ? Non. Si la réponse est non, proposer une correction de ce script pour qu'il effectue les conversions attendues.

Il faut corriger la 6^e ligne par « mettre E à $4984 * B$ » et la 12^e ligne par « mettre B à $E / 4984$ ».

Exercice 2 Forfait de photocopies Instruction conditionnelle

Une petite société implantée près d'une université propose des services de reprographie. La photocopie d'un document de format A4 (21 cm × 29,7 cm) coûte 8 centimes. Souhaitant fidéliser les étudiants et leur permettre d'obtenir des prix plus intéressants, à la première visite de l'étudiant, le responsable propose l'abonnement annuel suivant : l'étudiant paie 20 € pour réaliser un nombre maximum de 500 photocopies, puis 6 centimes pour chaque photocopie au-delà de la 500^e.

1 Compléter le tableau suivant.

Nombre N de copies	100	250	600
Prix sans abonnement (en €)	8	20	48
Prix avec abonnement (en €)	20	20	26

2 Exprimer le prix à payer P sans abonnement en fonction du nombre N de copies : $P = 0,08N$.

De quel type de fonction s'agit-il ? Il s'agit d'une fonction linéaire.

3 Exprimer le prix à payer P avec abonnement en fonction du nombre N de copies. On prendra soin de distinguer les cas $0 \leq N \leq 500$ et $N > 500$.

Si $0 \leq N \leq 500$ alors $P = 20$ et si $N > 500$ alors $P = 20 + (N - 500) \times 0,06$.

4 Le responsable souhaite concevoir un programme qui affiche directement le prix à payer par l'étudiant lors de sa première visite pour un nombre N de copies saisies. Il devra commencer par indiquer si l'étudiant est abonné (tarif 1) ou non (tarif 2). On note T la variable tarif ; au tarif 1, $T = 1$, au tarif 2, $T \neq 1$. Compléter l'algorithme ci-contre afin qu'il réponde au problème.

Langage naturel

Variables	T, N, P nombres entiers
Entrées	Saisir T Saisir N
Traitement	Si $T = 1$ Si $N \leq 500$ Affecter à P la valeur <u>20</u> Sinon Affecter à P la valeur <u>$20 + 0,06(N - 500)$</u> FinSi Sinon Affecter à P la valeur <u>$0,08N$</u> FinSi
Sortie	Afficher P

5 En utilisant les fiches sur l'intérieur de la couverture, traduire l'algorithme ci-dessus dans le langage de programmation de votre calculatrice (on pourra lui donner le nom : COPIES).

6 Exécuter le programme de la question précédente et vérifier ainsi les résultats obtenus en question 1.

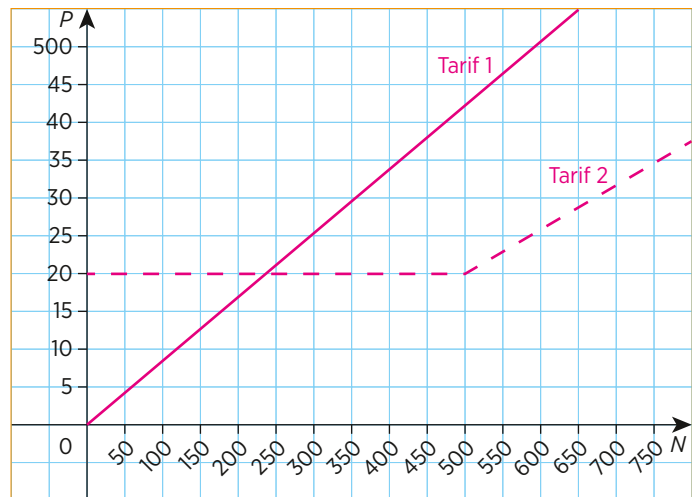
7 Représenter dans le repère ci-contre le prix à payer par l'étudiant lors de sa première visite en fonction du nombre N de copies. Tracer en bleu la courbe correspondant au tarif 1 et en rouge la courbe correspondant au tarif 2. On pourra s'aider de valeurs obtenues en exécutant le programme précédent.

```

T.I. Basic
Prompt T
Prompt N
If T=1
Then
If N≤500
Then
20→P
Else
20+0.06*(N-500)→P
End
Else
0.08*N→P
End
Disp P
  
```

```

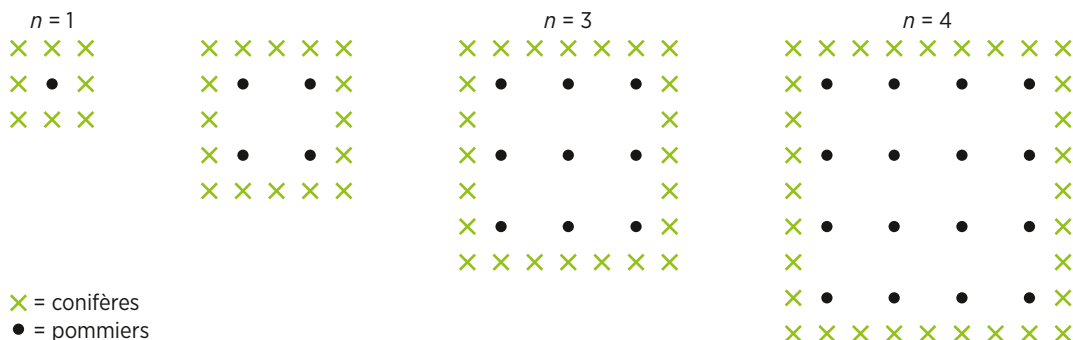
Basic Casio
?→T↓
?→N↓
If T=1↓
Then↓
If N≤500↓
Then 20→P↓
Else 20+0.06*(N-500)→P↓
IfEnd↓
Else 0.08*N→P↓
IfEnd↓
P↑
  
```



Exercice 3 Les pommiers Boucle non bornée

Un fermier plante des pommiers en carré. Afin de protéger ces arbres contre le vent, il plante des conifères tout autour du verger. Le schéma ci-dessous montre la disposition des pommiers et des conifères pour un nombre (n) de rangées de pommiers :

Source : Pisa 2000



On admettra ici que pour n rangées de pommiers, le nombre de pommiers est $P(n) = n^2$ et le nombre de conifères nécessaires pour les protéger est $C(n) = 8n$.

1 Proposer un algorithme qui, pour un nombre entier positif n de rangées saisies, calcule et affiche les nombres de pommiers P et de conifères nécessaires C correspondants.

Langage naturel

Variables	n, C, P nombres entiers
Entrée	Saisir n
Traitement	Affecter à P la valeur n^2 Affecter à C la valeur $8n$
Sortie	Afficher P Afficher C

2 Traduire cet algorithme dans le langage de programmation de votre choix.

T.I. Basic

```
Prompt N .....
N^2→P .....
8*N→C .....
Disp P .....
Disp C .....
```

Basic Casio

```
?→N↓ .....
N²→P↓ .....
8×N→C↓ .....
P▲ .....
C▲ .....
```

Python

```
n=int(input("n=")) .....
P=n**2 .....
C=8*n .....
print(P) .....
print(C) .....
```

3 En exécutant le programme précédent, compléter le tableau ci-dessous.

n	1	2	3	4	5	6
P	1	4	9	16	25	36
C	8	16	24	32	40	48

4 Par essais successifs, déterminer un nombre n de rangées de pommiers pour lequel $P(n) = C(n)$.

Pour $n = 7$, P vaut 49 et C vaut 56 ; pour $n = 8$, P vaut 64 et C vaut 64. $P(n) = C(n)$ pour $n = 8$.

5 Compléter l'algorithme ci-dessous de façon à ce qu'il détermine et affiche la solution de l'équation précédente.

Langage naturel

Variables	n, C, P nombres entiers
Initialisation	n prend la valeur 1 P prend la valeur 1 C prend la valeur 8
Traitement	Tant que $P \neq C$ Affecter à n la valeur $n + 1$ Affecter à P la valeur n^2 Affecter à C la valeur $8n$ FinTantQue
Sortie	Afficher n

Exercice 4

Sens de variation de la fonction carré

Boucle bornée

On considère l'algorithme suivant.

Langage naturel

Variables N, C et I nombres entiers
Initialisation N prend la valeur -3
Traitement Pour I allant de 1 à 5
 N prend la valeur $N + 1$
 Affecter à C la valeur N^2
 Afficher C
 FinPour
Fin de l'algorithme

1 Sans implanter le programme correspondant sur une machine, exécuter cet algorithme et compléter le tableau ci-dessous.

I		1	2	3	4	5
N	-3	-2	-1	0	1	2
C		4	1	0	1	4

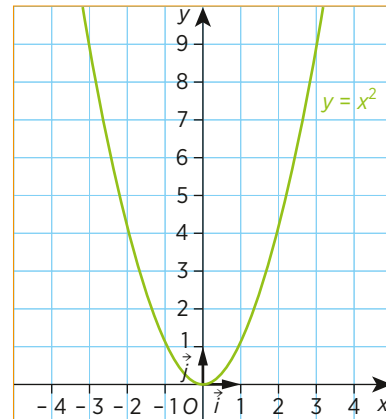
2 L'affirmation suivante est-elle exacte ? « Plus N est grand, plus N^2 est grand. »

L'affirmation est fausse : par exemple, pour $N = -2$, N^2 est plus grand que pour $N = 0$.

3 Quelle condition doit vérifier la valeur initiale de N pour que l'affirmation précédente soit toujours vraie ? Justifier la réponse en utilisant des résultats du cours sur la fonction de la variable x définie par $f : x \mapsto x^2$ ainsi que la courbe de cette fonction que l'on redonne ci-contre.

L'affirmation serait vraie si la fonction carré était croissante sur \mathbb{R} , or cette fonction est décroissante sur $]-\infty ; 0]$ et croissante sur $[0 ; +\infty[$.

La valeur initiale de N devrait donc être positive ou nulle.



4 Quel est le rôle de l'algorithme ci-contre ?

Cet algorithme sert à déterminer le plus petit entier

positif dont le carré est supérieur à 500 000.

Langage naturel

Variables N, C nombres entiers
Initialisation N prend la valeur 0
 C prend la valeur 0
Traitement Tant que $C < 500\,000$
 Affecter à N la valeur $N + 1$
 Affecter à C la valeur N^2
 Afficher C
 FinTantQue
Sortie Afficher N

5 Voici ci-contre les programmes en langage T.I. Basic et Basic Casio, traduisant l'algorithme précédent.

a) Implanter ce programme dans une calculatrice et l'exécuter (l'exécution est longue et les affichages sont nombreux).

T.I. Basic

```
PROGRAM: CARRE
:0→N
:0→C
:While C<500000
:N+1→N
:N^2→C
:Disp C
:End
:Disp N
```

Basic Casio

```

CARRE
0→N
0→C
While C<500000
N+1→N
N^2→C
C
TOP BOTTOM SEARCH MENU Aa CHAR
```

b) Quelle est la dernière valeur de C affichée ? 501 624.

c) Quelle est la valeur de N affichée ? 708.

En donner une interprétation : 708 est le plus petit entier positif dont le carré est supérieur à 500 000.

Exercice 5 Et de pas à pas... Boucles bornées et non bornées

On rappelle ici que la fonction f définie sur $] -\infty; 0[\cup] 0; +\infty[$ par $f(x) = \frac{1}{x}$ est décroissante sur chacun des intervalles $] -\infty; 0[$ et $] 0; +\infty[$.

Soit un entier n strictement positif ; on considère ici la somme $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$.

On a $S(1) = 1$; $S(2) = 1 + \frac{1}{2}$; $S(3) = 1 + \frac{1}{2} + \frac{1}{3}$, etc.

1 Sans utiliser d'algorithme, compléter le tableau suivant.

Valeur de n	1	2	3	4
Valeur exacte de $S(n)$ sous forme de fraction irréductible	1	$\frac{3}{2}$	$\frac{11}{6}$	$\frac{25}{12}$

2 Compléter la phrase suivante : Pour tout $n \geq 0$, plus n est grand, plus $S(n)$ est grand.....

3 Dans la suite de l'exercice, on cherche à savoir si l'on peut rendre $S(n)$ aussi grand que l'on veut pourvu que n soit « suffisamment grand ». Pour cela, on a écrit un algorithme qui, pour un entier N saisi, calcule $S(N)$. Des parties de cet algorithme ont été effacées ; retrouver les parties manquantes.

Pseudo-langage

```
S ← 0
Pour I allant de 1 à N
    S ← S + 1/I .....
FinPour
```

4 a) De la même manière, compléter le script ci-contre écrit en langage Python, traduisant l'algorithme précédent.

Python

```
S=0
N=int(input("Saisir N="))
for I in range(1,N+1):
    S=S+1/I .....
print (S)
```

b) Implanter ce script sur un ordinateur, puis l'exécuter et compléter le tableau ci-dessous avec les valeurs arrondies au millième.

n	10	50	75	100	150	200
$S(n)$	2,929	4,499	4,901	5,187	5,591	5,878

5 En observant le tableau de la question précédente, quel semble être le comportement de $S(n)$ lorsque n prend de « très grandes valeurs » ?

Lorsque n prend de très grandes valeurs, $S(n)$ semble augmenter de moins en moins rapidement.....

6 a) Écrire un algorithme qui, pour un nombre A saisi, détermine le plus petit entier N tel que $S(N) > A$. On utilisera la boucle « Tant que $S \leq A$ ».

b) Traduire, puis implanter le programme correspondant en langage Python et en l'exécutant, déterminer le plus petit entier N tel que $S(N) > 6$: $N = \dots 227 \dots$.

Langage naturel

Variables	S, N, A nombres flottants
Initialisation	S prend la valeur 0 N prend la valeur 0
Entrée	Saisir A
Traitement	Tant que $S \leq A$ N prend la valeur $N + 1$ S prend la valeur $S + 1/N$ FinTantQue
Sortie	Afficher N

Python

```
S=0 .....
N=0 .....
A=float(input("A=")) .....
while S<=A: .....
    N=N+1 .....
    S=S+1/N .....
print(N) .....
```



Voitures autonomes et sécurité routière

Boucle bornée

« **Voitures autonomes : la Californie autorise les tests sans chauffeur**
 Dans la course à la voiture autonome, les pouvoirs publics ne sont pas en reste – États, villes ou régions rivalisant pour accueillir les équipes d'ingénieurs travaillant sur la conduite automatique. L'État de Californie vient de frapper fort dans cette partie, en assouplissant le cadre des tests de voitures robots menés sur son territoire. Celles-ci vont pouvoir dès l'an prochain évoluer sur la route en l'absence de chauffeur à bord.
Les Echos – octobre 2017



PARTIE 1 Un algorithme pour évaluer les distances de freinage

Si un véhicule autonome roule à une vitesse V , exprimée en km/h, dès que le système de frein est actionné, la distance parcourue par le véhicule jusqu'à son arrêt total, en mètres, est donnée par : $D_f = \frac{V^2}{254 \times f}$ où f est le coefficient d'adhérence, qui dépend de l'état de la chaussée.

Par la suite, on prendra pour valeurs de f , sur route sèche : $f = 0,8$ et sur route mouillée : $f = 0,4$.

1 On cherche à élaborer un algorithme qui, une fois le coefficient d'adhérence f et la vitesse V saisis, calcule et affiche la distance de freinage D_f .

a) On crée ici trois variables :

- F : le coefficient d'adhérence
- V : la vitesse de la voiture
- D_f : la distance de freinage

b) Compléter l'algorithme suivant afin qu'il réponde au problème.

c) Traduire cet algorithme dans le langage de programmation Python.

Langage naturel

Variables	F, V et D_f nombres flottants
Entrées	Saisir F Saisir V
Traitement	Affecter à D_f la valeur $\frac{V^2}{254 \times f}$
Sortie	Afficher D_f

Python

```
F=float(input("F="))
V=float(input("V="))
Df=V**2/(254*F)
print(Df)
```

2 En exécutant le programme précédent, compléter le tableau ci-dessous.

Vitesse (km/h)	50	100	120
Distance de freinage (en m) avec le coefficient d'adhérence 0,8	12,30	49,21	70,87
Distance de freinage (en m) avec le coefficient d'adhérence 0,4	24,61	98,42	141,73

3 a) Compléter la phrase : à chaque valeur de F on définit ainsi une fonction de la variable V . On peut noter

$$D_{0,8}(V) = \frac{V^2}{203,2} \text{ et } D_{0,4}(V) = \frac{V^2}{101,6}$$

b) On considère ici que les véhicules autonomes seront limités à la vitesse de 130 km/h.

Préciser l'ensemble de définition ainsi que le sens de variation de chacune des fonctions $D_{0,8}$ et $D_{0,4}$.

La fonction $D_{0,8}$ a pour ensemble de définition : $I_{0,8} = [0 ; 130]$ et cette fonction est croissante.

La fonction $D_{0,4}$ a pour ensemble de définition : $I_{0,4} = [0 ; 130]$ et cette fonction est croissante.

PARTIE 2 Distance de sécurité des véhicules autonomes

Lorsque deux véhicules se suivent, le code de la route indique que le second véhicule doit laisser une distance égale à la distance parcourue par son véhicule pendant une durée d'au moins 2 secondes. Cette distance est appelée distance de sécurité.

- 4 Montrer que la distance de sécurité à respecter par un véhicule autonome (en m) est égale à $\frac{5}{9}$ de la vitesse (en km/h).

$$V = \frac{Ds}{T}; Ds = V \times T; \text{ pour } T = 2 \text{ secondes, } Ds = \frac{V}{3,6} \times 2 = \frac{5}{9} \times V.$$

- 5 En déduire que cette distance de sécurité Ds (en m) est une fonction de la variable vitesse V (en km/h).

$$\text{On peut écrire } Ds = \frac{5}{9}V.$$

De quel type est cette fonction ? C'est une fonction linéaire. Quel est son sens de variation ? Elle est croissante.

- 6 On considère l'algorithme ci-contre. Sans traduire cet algorithme en programme, exécuter cet algorithme puis compléter le tableau ci-dessous.

I	1	2	3	4	5	6
V	20	40	60	80	100	120
Ds	11,1	22,2	33,3	44,4	55,5	66,7

Langage naturel

Variables	V, Ds nombres flottants
Initialisation	V prend la valeur 0
Traitement	Pour I allant de 1 à 6 V prend la valeur $V + 20$ Ds prend la valeur $\frac{5}{9}V$
	FinPour
Sortie	Afficher Ds

- 7 Après avoir traduit cet algorithme dans le langage de programmation Python, implanter ce programme et l'exécuter pour vérifier les résultats consignés dans le tableau précédent.

Python

```
V=0
for I in range(1,7):
    V=V+20
    Ds=5/9*V
print(Ds)
```

PARTIE 3 Adaptation de la vitesse d'un véhicule autonome

En réalité, un véhicule autonome doit adapter sa vitesse V (en km/h) à la distance D qui le sépare du véhicule qui le précède (en m), sachant que ce véhicule peut ne pas rouler à vitesse constante. On suppose ici que le véhicule autonome ne peut dépasser aucun autre véhicule. On reprend ici la formule de la partie 2 : $Ds = \frac{5}{9}V$.

- 8 Proposer un algorithme qui calcule et affiche les vitesses maximales correspondant à toutes les distances entières (en m) comprises entre 0 et 50.

Langage naturel

Variables :	V, Ds
Initialisation :	Ds prend la valeur 0
Traitement :	Pour I allant de 1 à 50
	Ds prend la valeur $Ds + 1$
	V prend la valeur $\frac{9}{5}Ds$
	FinPour
Sortie :	Afficher V

Fonction polynôme de degré 2

Exercice 1 **Forme canonique** Instruction conditionnelle

On considère les deux algorithmes ci-dessous.

Algorithme 1 – Langage naturel

Entrée	Saisir X
Traitement	Affecter à Y la valeur $X^2 - 2X$
Sortie	Afficher Y

Algorithme 2 – Langage naturel

Entrée	Saisir X
Traitement	Affecter à Y la valeur $X - 1$ Affecter à Y la valeur $Y^2 - 1$
Sortie	Afficher Y

1 Quelles sont les variables de ces algorithmes ? De quel type de variable peut-il s'agir ?

Les variables de ces algorithmes sont X et Y , ce sont des nombres flottants.

2 Exécuter ces deux algorithmes lorsque la valeur de X est 4. Quelles sont les valeurs de Y ?

Pour l'algorithme 1 : $4^2 - 2 \times 4 = 8$, donc $Y = 8$.

Pour l'algorithme 2 : $(4 - 1)^2 - 1 = 8$, donc $Y = 8$.

3 Reprendre la question 2 pour $X = -1$.

Pour l'algorithme 1 : $(-1)^2 - 2 \times (-1) = 3$, donc $Y = 3$.

Pour l'algorithme 2 : $(-1 - 1)^2 - 1 = 3$, donc $Y = 3$.

4 Quelle conjecture peut-on émettre sur les valeurs obtenues par ces deux algorithmes ? Justifier.

On peut conjecturer que les valeurs obtenues par ces deux algorithmes sont égales.

Justification : pour tout réel X , $(X - 1)^2 - 1 = X^2 - 2X + 1 - 1 = X^2 - 2X$.

5 Traduire l'algorithme 2 dans le langage de programmation de votre calculatrice ou en Python, puis implanter ce programme sur une machine. L'exécuter et vérifier les valeurs obtenues dans les questions 2 et 3.

T.I. Basic

```
Prompt X
X-1→Y
Y^2-1→Y
Disp Y
```

Basic Casio

```
"Saisir X"?→X
X-1→Y
Y^2-1→Y
Y
```

Python

```
print("Entrer X :")
X=float(input())
Y=X-1
Y=Y*Y-1
print("Y=",Y)
```

6 Si la valeur de la variable Y obtenue après l'exécution de l'algorithme 2 est 15, quelles sont les valeurs possibles de X avant l'exécution de cet algorithme ?

$(X - 1)^2 - 1 = 15$ équivaut à $(X - 1)^2 = 16$ ou encore $X - 1 = 4$ ou $X - 1 = -4$, c'est-à-dire $X = 5$ ou $X = -3$.

7 Y est un nombre réel. L'objectif de cette question est d'écrire un programme qui calcule les éventuelles valeurs de X vérifiant : $(X - 1)^2 - 1 = Y$, c'est-à-dire un programme qui « remonte » les programmes associés à l'algorithme 2.

a) Pour quelles valeurs de Y n'y a-t-il pas de valeur possible pour X ? Une unique valeur possible ? Deux valeurs possibles ?

Pour $Y < -1$: pas de valeur possible.

Pour $Y = -1$: une unique valeur possible : 1.

Pour $Y > -1$: deux valeurs possibles.

b) Compléter l'extrait de programme en Python ci-contre.

c) Dans le cas où $Y > -1$, justifier que les valeurs cherchées sont $\sqrt{Y+1}+1$ et $-\sqrt{Y+1}+1$.

$(X-1)^2 - 1 = X(X-2)$. On remplace par $\sqrt{Y+1}+1$:
 $(\sqrt{Y+1}+1) \times (\sqrt{Y+1}+1-2) = \sqrt{Y+1}-1+1+Y-\sqrt{Y+1}+1 = Y$.
 De même, $(-\sqrt{Y+1}+1) \times (-\sqrt{Y+1}+1-2) = Y$.

Python

```
Y=float(input("Y="))
if Y<-1 :
    print("pas de valeur possible")
if Y == -1 :
    print("une unique valeur possible : 1")
if Y>-1 :
    print("deux valeurs possibles", X1, X2)
```

d) On complète le programme précédent par les instructions ci-contre. Implanter sur un ordinateur ce programme et le tester pour $Y = -2$, $Y = -1$, $Y = 0$ et $Y = 1$.

Python

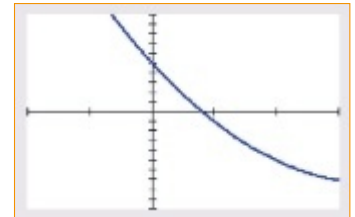
```
X1=(Y+1)**0.5+1
X2=- (Y+1)**0.5+1
```

On obtient [pas de valeur possible], [une unique valeur possible : 1], [deux valeurs possibles : 2, 0],
 [deux valeurs possibles : $\sqrt{2}+1, 1-\sqrt{2}$].

Exercice 2 Algorithme de dichotomie

Instruction conditionnelle et boucle non bornée

On considère la fonction f définie sur l'intervalle $[-2, 3]$ par $f(x) = x^2 - 7x + 5$. On a tracé ci-contre, à l'aide d'une calculatrice, la courbe C_f représentative de la fonction f . Une graduation correspond à une unité.



1 Encadrer entre deux entiers la valeur x_0 solution de l'équation $f(x) = 0$.

x_0 est encadré entre l'entier 0 et 1. En effet, $x_0 \in [0; 1]$.

2 a) On remarquera que $f(0) \times f(1) < 0$. Calculer $f(0) \times f\left(\frac{1}{2}\right)$ et $f\left(\frac{1}{2}\right) \times f(1)$.

$f(0) \times f\left(\frac{1}{2}\right) = 5 \times 1,75 = 8,75$. $f\left(\frac{1}{2}\right) \times f(1) = 1,75 \times (-1) = -1,75$.

b) En déduire si x_0 appartient à l'intervalle $\left[0; \frac{1}{2}\right]$ ou $\left[\frac{1}{2}; 1\right]$.

$x_0 \in \left[\frac{1}{2}; 1\right]$, puisque $f\left(\frac{1}{2}\right) \times f(1) < 0$.

3 On donne ci-dessous deux copies d'écran de calculatrices et une copie d'un script en langage Python. Ces trois programmes correspondent à un seul algorithme appelé « algorithme de dichotomie ».

T.I. Basic

```
:Prompt A,B,E
:While (B-A)>E
:(A+B)/2→M
:If Y1(A)*Y1(M)≤0
:Then
:M→B
:Else
:M→A
:End
:End
:Disp A,B
```

Basic Casio

```
DICHOTOM
?→A
?→B
?→E
While (B-A)>E
(A+B)÷2→M
M→X:Y1→I
A→X:Y1→J
If I×J≤0
Then M→B
Else M→A
IfEnd
WhileEnd
A
B
```

Python

```
def f(x):
    return x*x-7*x+5
a=float(input("Entrez la borne entière à gauche a="))
b=float(input("Entrez la borne entière à droite b="))
e=float(input("Entrez l'amplitude e="))
while (b-a)>e :
    m=(a+b)/2
    if f(a)*f(m)<=0 :
        b=m
    else :
        a=m
print("x0 est compris entre ",a," et ",b)
```

a) Faire fonctionner cet algorithme sans machine en complétant le tableau ci-dessous avec $a = 0$, $b = 1$ et $e = 0,1$.

	a	b	Condition $(b - a) > e$	m	Signe de $f(a) \times f(m)$
Avant la 1 ^{re} condition	0	1	Vrai puisque $(1 - 0) = 1 > 0,1$	0,5	Positif
Avant la 2 ^e condition	0,5	1	Vrai puisque $(1 - 0,5) = 0,5 > 0,1$	0,75	Positif
Avant la 3 ^e condition	0,75	1	Vrai puisque $(1 - 0,75) = 0,25 > 0,1$	0,875	Négatif
Avant la 4 ^e condition	0,75	0,875	Vrai puisque $(0,875 - 0,75) = 0,125 > 0,1$	0,8125	Négatif
Avant la 5 ^e condition	0,75	0,8125	Faux puisque $(0,8125 - 0,75) = 0,0625 \leq 0,1$	×	×

b) Quelles sont les valeurs affichées en sortie ? En déduire un encadrement de x_0 .

Les valeurs affichées par l'algorithme sont 0,75 et 0,8125. On peut en déduire que $0,75 \leq x_0 \leq 0,8125$.

4 a) Programmer cet algorithme sur la calculatrice ou sur un ordinateur.

b) Utiliser ce programme pour donner un encadrement de x_0 d'amplitude $e = 0,001$ (on arrondira les résultats à 10^{-3} près).

On donne en entrée $a = 0$, $b = 1$ et $e = 0,001$. On obtient $0,807 \leq x_0 \leq 0,808$.

c) En déduire une valeur approchée de x_0 à 10^{-2} près.

On en déduit que $x_0 \approx 0,81$.

5 Déterminer à l'aide de la calculatrice puis de l'algorithme une valeur approchée à 10^{-2} près de la deuxième solution x_1 de l'équation $f(x) = 0$ sur \mathbb{R} .

À l'aide de la calculatrice, on constate que la deuxième solution est comprise entre 6 et 7.

Il suffit d'utiliser l'algorithme avec $a = 6$, $b = 7$ et $e = 0,001$.

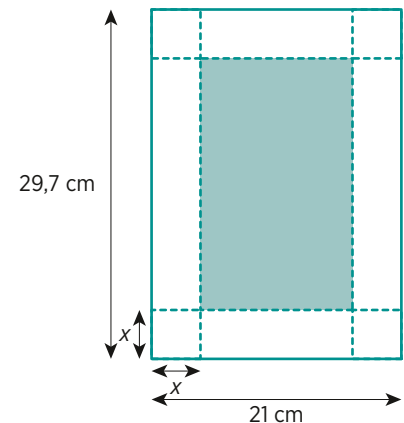
On obtient l'encadrement suivant : $6,192 \leq x_0 \leq 6,193$.

On en conclut que $x_0 \approx 6,19$.

Exercice 3 Problème de modélisation

Instruction conditionnelle et boucle non bornée

Une éditrice de bande dessinée veut se décider sur la taille des marges des pages en A4 de leur prochain best-seller. En effet, elle souhaite laisser 360 cm^2 comme superficie de zone d'impression. Une page A4 a une dimension de $21,0 \times 29,7 \text{ cm}$. On note x la taille de la marge horizontale et verticale. On définit la fonction f qui à x associe l'aire de la zone d'impression.



1 a) Dans quel intervalle x peut-il varier ?

x peut varier dans l'intervalle $[0; 10,5]$.

b) Montrer que l'expression de $f(x)$ peut s'écrire $f(x) = (21 - 2x)(29,7 - 2x)$.

Le rectangle formé par la zone d'impression est de taille $l = 21 - 2x$ et $L = 29,7 - 2x$.

Or l'aire d'un rectangle est $l \times L = (21 - 2x)(29,7 - 2x)$.

Donc $f(x) = (21 - 2x)(29,7 - 2x)$.

2 Montrer que $f(x) = -4x^2 + 101,4x + 623,7$.

$f(x) = (21 - 2x)(29,7 - 2x)$

$= 21 \times 29,7 + 21 \times (-2x) - 2x \times 29,7 - 2x \times (-2x)$

$= 4x^2 - 101,4x + 623,7$.

3 On souhaite résoudre l'équation $f(x) = 360$.

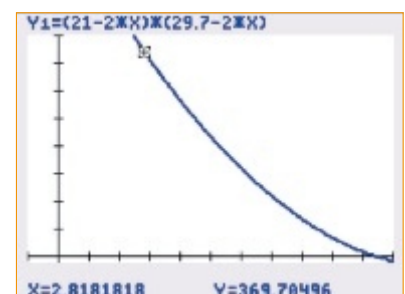
a) À l'aide de la calculatrice, on a tracé la courbe représentative de la fonction. En abscisse 1 unité correspond à 1 cm. En ordonnée 1 unité correspond à 50 cm^2 .

Quel est le sens de variation de la fonction sur $[0; 10,5]$?

La fonction f est strictement décroissante sur $[0; 10,5]$.

b) Justifier qu'il y a une seule solution à l'équation $f(x) = 360$.

La fonction est strictement décroissante donc il y a au plus une solution.



c) On note x_0 cette solution. Déterminer deux entiers a et b tels que $a < x_0 < b$.

On peut remarquer que $2 < x_0 < 3$. Donc $a = 2$ et $b = 3$.

4 On a écrit l'algorithme en Python ci-contre.

Python

```
def f(x):
    return 4*x**2-101.4*x+623.7

x=int(input("Entrer un entier initial de recherche : "))
while f(x)>=360:
    x=x+0.01
print(x)
```

a) Que fait cet algorithme ?

Cet algorithme lance une recherche pas à pas de x_0 avec une approximation à 10^{-2} près.

b) Que se passe-t-il lorsqu'on entre 3 comme valeur initiale de recherche ? L'algorithme ne s'arrête pas.

c) Implanter ce programme et déterminer une valeur approchée de x_0 à 10^{-2} près. $x_0 \approx 2,95$.

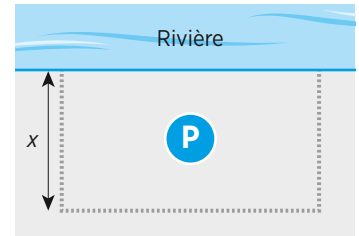
5 Modifier le programme afin qu'il teste si la valeur entrée par l'utilisateur est correcte.

On ajoute à la fin du programme précédent : else:

print(« Vous avez entré un entier trop grand. »)

Exercice 4 Un problème d'optimisation (Notion de fonction)

On dispose de 300 m de clôture (portail compris) pour délimiter une zone de stationnement rectangulaire. Comme l'un des côtés de cette zone est le bord d'une rivière, seuls trois côtés seront clôturés. Des contraintes de réalisation imposent que les longueurs des côtés de cette zone soient entières (en m). On cherche à déterminer l'aire maximale de cette zone de stationnement.



1 Exprimer l'aire de cette zone de stationnement en fonction de x .

$$A(x) = x(300 - 2x) = -2x^2 + 300x$$

2 Compléter le script en Python ci-contre qui définit la fonction A (au sens algorithmique).

Python

```
def A(x):
    return -2*x**2+300*x
```

3 a) Quelles sont les valeurs possibles pour x ?

Les valeurs de x sont les entiers de l'intervalle $[0 ; 150]$.

b) On admet que, lorsque x augmente, cette fonction est « *croissante puis décroissante* ».

Le principe de l'algorithme ci-contre est de calculer les images $A(x)$ successives, pour des valeurs croissantes de x , tant que ces images augmentent. L'algorithme s'arrête lorsque les images diminuent et affiche la valeur maximale de $A(x)$. Compléter cet algorithme.

Langage naturel

Tant que $A(x+1) > A(x)$
 x prend la valeur $x+1$
 Fin Tant que
 Afficher $A(x)$

4 Implanter cet algorithme sur la calculatrice ou un ordinateur et donner l'aire maximale de la zone de stationnement.

Python

```
def A(x):
    return -2*x**2+300*x

x=float()
x=0
while A(x+1)>A(x):
    x=x+1
print A(x)
```

T.I. Basic

```
0 → X
While -2(X+1)2+300(X+1)>-2X2+300X
X+1 → X
End
Disp -2X2+300X
```

Basic Casio

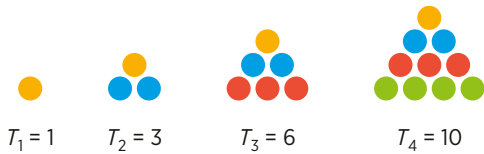
```
0 → X
While -2(X+1)2+300(X+1)>-2X2+300X ↓
X+1 → X ↓
WhileEnd ↓
-2X2+300X ▲
```



Nombres triangulaires

Notion de fonction

Des mathématiciens grecs (probablement de l'école de Pythagore, VI^e-V^e siècles avant J.-C.) représentaient certains nombres à l'aide de polygones (nombres polygonaux). Les nombres triangulaires en sont des cas particuliers. Voici deux représentations des quatre premiers nombres triangulaires :



PARTIE 1 Un script qui calcule les premiers nombres triangulaires

1 Calculer T_5, T_6, T_7, T_{10} .

$T_5 = 15$ $T_6 = 21$ $T_7 = 28$ $T_{10} = 55$

2 Compléter :

- a) Pour calculer T_5 , on ajoute 5 à T_4 .
- b) Pour calculer T_{n+1} , on ajoute $n+1$ à T_n .
- c) Compléter l'algorithme ci-dessous dont le rôle est de calculer T_{20} .

Langage naturel

Variable	T de type nombre entier
Initialisation	T prend la valeur <u>1</u>
Traitement	Pour i allant de 2 à <u>20</u> T prend la valeur <u>$T+i$</u>
	FinPour
Sortie	Afficher T

3 Modifier le script suivant en langage Python pour qu'il affiche toutes les valeurs de T_2 à T_{20} et l'implanter sur un ordinateur, puis l'exécuter.

Python

```
T=int()
T=1
for i in range(2,21):
    T=T+i
print(T)
```

Python

```
T=int()
T=1
for i in range(2,21):
    T=T+i
print(T)
```

4 Modifier le script pour qu'il affiche la valeur de T_n , la valeur de n supérieure ou égale à 2 étant donnée par l'utilisateur. L'implanter sur un ordinateur et l'exécuter.

Python

```
n=int(input("n="))
T=int()
i=int()
i=2
T=1
while i<=n:
    T=T+i
    i=i+1
print(T)
```

PARTIE 2 Utilisation d'une fonction

5 Pour calculer un nombre triangulaire en évitant de calculer ceux qui le précèdent, on veut définir une fonction (au sens algorithmique) d'un argument : le nombre n .

a) En utilisant cet assemblage de deux nombres triangulaires identiques, conjecturer une expression de T_n en fonction de n .

$$T_n = \frac{n(n+1)}{2}$$

Compléter : $T_n = \dots 0,5 \dots n^2 + \dots 0,5 \dots n$.

b) Compléter le script ci-contre qui renvoie le nombre T_n , en le définissant par une fonction de la variable n .

c) Compléter le script ci-contre dont le rôle est de calculer un nombre triangulaire, la valeur de n étant donnée par l'utilisateur, puis l'implanter sur un ordinateur et l'exécuter.

6 Voici ci-dessous un script pour tester si un nombre entier strictement positif, donné par l'utilisateur, est triangulaire.

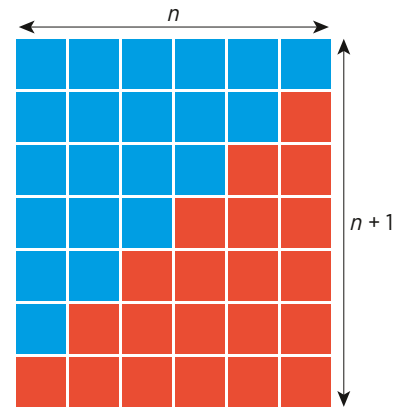
Python

```
def nombre(n):
    T=0.5*n**2+0.5*n
    return T
P=int(input("nombre="))
n=int()
T=int()
n=1
T=1
while T<P:
    n=n+1
    T=nombre(n)
    if T==P:
        print("Ce nombre est triangulaire avec n=", n)
    elif T>P:
        print("Ce nombre n'est pas triangulaire")
```

a) Faire fonctionner cet algorithme sans machine en complétant le tableau ci-dessous avec $P = 16$. Quel est le message en sortie de l'algorithme ?

	n	T	Condition $T < P$
Initialisation	1	1	Vrai puisque $1 < 16$
Après la 1 ^{re} boucle	2	3	Vrai puisque $3 < 16$
Après la 3 ^e boucle	3	6	Vrai puisque $6 < 16$
Après la 4 ^e boucle	4	10	Vrai puisque $10 < 16$
Après la 5 ^e boucle	5	15	Vrai puisque $15 < 16$
Après la 6 ^e boucle	6	21	Faux puisque $21 > 16$

Puisque $T \neq P$, l'algorithme affiche « Ce nombre n'est pas triangulaire ».



Python

```
def nombre(n):
    T=0.5*n**2+0.5*n
    return T
```

Python

```
def nombre(n):
    T=0.5*n**2+0.5*n
    return T
n=int(input("n="))
nombre(n)
print(T)
```

b) Implanter ce programme sur un ordinateur. Vérifier que le nombre 2016 est triangulaire avec $n = 63$.

En exécutant l'algorithme précédent, on trouve que $2016 = T_{63}$.

c) Vérifier que les nombres 2017 et 2018 ne sont pas triangulaires.

En exécutant l'algorithme, on remarque que 2017 et 2018 ne sont pas triangulaires.

d) Quelle sera la prochaine « année triangulaire » ?

$T_{64} = 2080$.

Exercice 1 Polygones réguliers

Boucle bornée


On considère le script ci-contre.

```

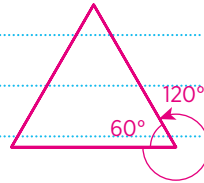
Scratch
quand est cliqué
effacer tout
cacher
aller à x: -70 y: -70
s'orienter à 90°
style en position d'écriture
avancer de 20
tourner de 120 degrés
avancer de 20
tourner de 120 degrés
avancer de 20
    
```

1 a) Implanter et exécuter ce script sur un ordinateur. Quel est le rôle de ce script ?

Ce script permet le tracé d'un triangle équilatéral de côté 20 (pixels).

b) Expliquer la commande :  à l'aide d'un schéma.

Chaque angle du triangle équilatéral mesure 60°, comme le premier côté tracé est parallèle à l'axe des abscisses, il faut tourner de $180 - 60 = 120^\circ$.



2 Modifier le script précédent pour tracer un carré de côté 20.

Il faut changer les deux lignes avec les valeurs d'angle en remplaçant 120 par 90, puis ajouter les deux instructions « tourner de 90 degrés » et « avancer de 20 » une quatrième fois.

3 On admet dans cette question que pour tracer un polygone régulier à n côtés de longueurs 20, les instructions sont celles ci-contre.

```

répéter n fois
avancer de 20
tourner de 360 / n degrés
    
```

a) Modifier le script précédent de façon à ce qu'il demande le nombre n de côtés à l'utilisateur avant de procéder au tracé du polygone régulier à n côtés.

Après « stylo en position d'écriture » il faut supprimer les instructions, ajouter l'instruction « demander nombre de côtés du polygone et attendre » puis « à n attribuer réponse » avant de placer la boucle bornée.

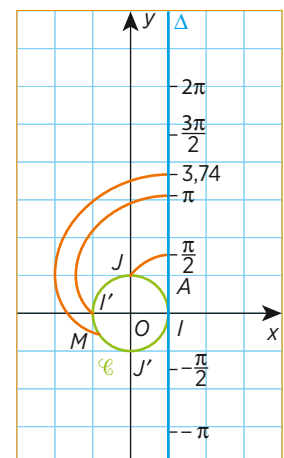
b) Tracer un polygone régulier à 36 côtés avec ce script. Quelle est la mesure en degrés de chacun des angles de ce polygone ?

Chacun des angles de ce polygone mesure : $180 - 10 = 170^\circ$.


Exercice 2 Points images sur le cercle trigonométrique

Instruction conditionnelle

Dans le plan muni du repère orthonormé (O, I, J) on note \mathcal{C} le cercle trigonométrique. A est le point de coordonnées $(1, 1)$ et Δ la droite (IA) munie du repère (I, A) . Cette droite, dite « droite des réels », s'enroule autour du cercle \mathcal{C} .



1 a) Placer sur le cercle trigonométrique ci-après les points M, N, P et Q images respectives des nombres réels suivants : $\frac{\pi}{3}, \frac{4\pi}{3}, \frac{13\pi}{3}, -\frac{2\pi}{3}$.

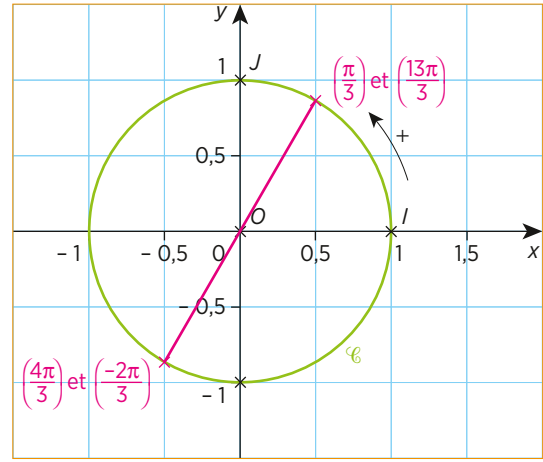
 **BESOIN D'AIDE ?**

$\frac{\pi}{3}$ correspond à un angle de 60° ; pour placer les points, on peut commencer par construire un triangle équilatéral.

b) Que constate-t-on ? Expliquer votre réponse.

Les nombres $\frac{\pi}{3}$ et $\frac{13\pi}{3}$ ont le même point image sur \mathbb{C} .

Les nombres $\frac{4\pi}{3}$ et $\frac{-2\pi}{3}$ ont le même point image sur \mathbb{C} .



2 On considère l'algorithme ci-contre.

a) Exécuter cet algorithme avec $x = \frac{\pi}{3}$ et $y = \frac{13\pi}{3}$.

$d = \frac{13\pi - \pi}{3 - \pi} = 4$. Affichage : **confondus**.

b) Exécuter cet algorithme avec $x = \frac{\pi}{4}$ et $y = \frac{1054\pi}{4}$.

$d = \frac{1054\pi - \pi}{4 - \pi} = 263,25$. Affichage : **distincts**.

Langage naturel

Variables	x, y, d nombres réels
Entrées	Saisir x Saisir y
Traitement	Affecter à d la valeur $\frac{y-x}{\pi}$ Si d est multiple de 2 Afficher « confondus » Sinon Afficher « distincts » FinSi
Fin de l'algorithme	

c) Pour tester si un nombre N est multiple de 2 à la calculatrice, on utilise la condition :

T.I. Basic

```
: If partEnt(N/2)=N/2
```

Basic Casio

```
If Int (N÷2)=N÷2↵
```

Écrire et implanter sur une calculatrice le programme correspondant à l'algorithme précédent et le tester avec les valeurs données en 2.a) et 2.b).

T.I. Basic

```
Prompt X
Prompt Y
(Y-X)/π→D
If partEnt(D/2)=D/2
Then
Disp "CONFONDUS"
Else
Disp "DISTINCTS"
End
```

Basic Casio

```
?→X↵
?→Y↵
(Y-X)÷π→D↵
If Int (D/2)=D÷2↵
Then "CONFONDUS"↵
Else "DISTINCTS"↵
IfEnd↵
```

Exercice 3

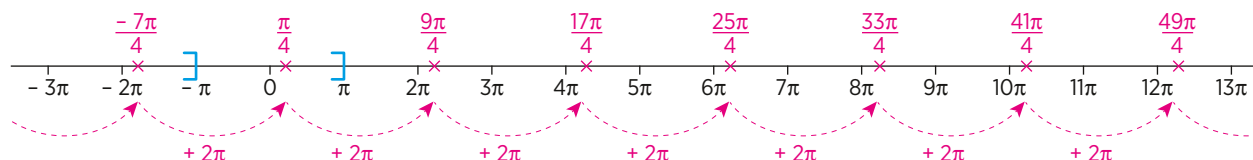
Recherche de la valeur de la mesure « principale »

Boucle non bornée

On s'intéresse ici aux nombres ayant le même point image sur le cercle trigonométrique.

1 Justifier que le nombre $\frac{17\pi}{4}$ a le même point image que le nombre $\frac{25\pi}{4}$: $\frac{17\pi}{4} + 2\pi = \frac{25\pi}{4}$

Déterminer d'autres nombres ayant le même point image que les deux nombres précédents.



2 On admettra ici que pour tout nombre réel x donné, il existe un et un seul nombre x_0 appartenant à l'intervalle $]-\pi; \pi]$ ayant le même point image que le nombre x .

a) Déterminer x_0 lorsque $x = \frac{17\pi}{4}$. $x_0 = \frac{17\pi}{4} - 4\pi = \frac{17\pi}{4} - \frac{16\pi}{4} = \frac{\pi}{4}$

b) Déterminer x_0 lorsque $x = -\frac{5\pi}{3}$. $x_0 = -\frac{5\pi}{3} + 2\pi = -\frac{5\pi}{3} + \frac{6\pi}{3} = \frac{\pi}{3}$

c) Compléter l'algorithme ci-contre afin qu'il détermine x_0 , une fois la valeur de x saisie par l'utilisateur.

Langage naturel

Variable	X nombre flottant
Entrée	Saisir X
Traitement	Tant que $X > \dots \pi \dots$ Affecter à X la valeur $X - \dots 2\pi \dots$ FinTantQue Tant que $X \leq \dots -\pi \dots$ Affecter à X la valeur $X + \dots 2\pi \dots$ FinTantQue
Sortie	Afficher $\dots X \dots$

Exercice 4

Déterminer « la » valeur de $\sin(x)$ connaissant $\cos(x)$

Instruction conditionnelle

Dans cet exercice, x désigne un nombre réel appartenant à l'intervalle $]-\pi; \pi]$.

1 a) Quel est le signe de $\sin(x)$ selon les valeurs de x ?

Pour $x \in]-\pi; 0[$ $\sin x < 0$ et pour $x \in [0; \pi]$ $\sin x \geq 0$.

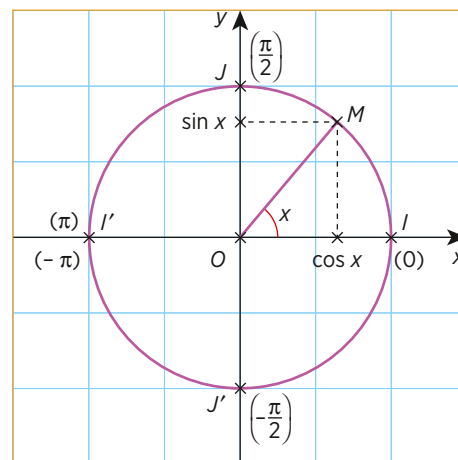
b) Lorsque $\cos(x) = \frac{3}{4}$, quelles sont les valeurs possibles de $\sin(x)$?

Pour tout x on a $\cos^2 x + \sin^2 x = 1$. Si $\cos x = \frac{3}{4}$, $\sin^2 x = 1 - \left(\frac{3}{4}\right)^2 = \frac{7}{16}$
on a donc $\sin x = \pm \sqrt{\frac{7}{16}} = \pm \frac{\sqrt{7}}{4}$.

c) Lorsque $\cos(x) = -\frac{2}{5}$, quelles sont les valeurs possibles de $\sin(x)$?

$\sin^2 x = \frac{21}{25}$ d'où $\sin x = \pm \frac{\sqrt{21}}{5}$.

d) Peut-on avoir $\cos(x) = \frac{5}{2}$? Justifier. Non. Pour tout x on a $-1 \leq \cos x \leq 1$ et $-1 \leq \sin x \leq 1$.



- 2 Karima propose le programme suivant écrit en langage Python. Elle pense que ce programme, une fois implanté, permet, pour une valeur de $\cos(x)$ saisie, à l'aide de la question posée, de faire afficher l'unique valeur possible de $\sin(x)$ correspondante. Ce programme comporte des erreurs ; repérer et corriger ces erreurs.

```
from math import *
cos=float(input(" Pour x appartenant à l'intervalle ]-pi,pi], saisir cos(x)="))
if cos>=-1 or cos<=1:
    reponse=input("x appartient-il à l'intervalle ]0,pi] ? Repondre par OUI ou NON ")
    if reponse=="OUI":
        sin=sqrt(1-cos*cos)
    else:
        sin=sqrt(cos*cos-1)
print(sin)
```

Il faut remplacer le « or » à la ligne 3 par « and » et la formule de calcul du sinus après le « else » par « -sqrt(1-cos*cos) ».

Exercice 5 Boucle bornée Une somme comportant un grand nombre de termes

Lors d'un test de mathématiques pour lequel la calculatrice est autorisée, Jules et Hana doivent calculer la somme suivante : $S = 1 + \cos \frac{\pi}{4} + \cos \frac{2\pi}{4} + \cos \frac{3\pi}{4} + \cos \frac{4\pi}{4} + \cos \frac{5\pi}{4} + \dots + \cos \frac{100\pi}{4}$.
Ils pensent que la saisie de cette somme à la machine leur prendra beaucoup de temps et cherchent un moyen de trouver rapidement la réponse.

- 1 En observant attentivement les termes de cette somme, trouver une expression générale pour chacun de ces termes et compter le nombre de termes.

Chaque terme s'écrit $\cos\left(\frac{n\pi}{4}\right)$ avec n entier compris entre 0 et 100. Cette somme comporte 101 termes.

- 2 Jules et Hana proposent les algorithmes ci-contre en pseudo-langage.

- a) Chacun de ces algorithmes répond-il au problème ?

Les deux algorithmes répondent au problème. Jules

considère le premier terme comme $\cos\left(\frac{0\pi}{4}\right) = \cos 0 = 1$

alors que Hana le distingue des autres.

- b) Écrire un programme répondant au problème dans le langage de votre calculatrice. On pensera à faire afficher en sortie la valeur de la variable S.

- c) Exécuter le programme précédent et en déduire la valeur de S. À l'affichage on obtient $S = 0$.

- d) Jules et Hana auraient-ils pu calculer rapidement la valeur de S si la calculatrice n'avait pas été autorisée pour ce devoir ? Justifier la réponse.

$$S = 1 + \cos \frac{\pi}{4} + \cos \frac{2\pi}{4} + \cos \frac{3\pi}{4} + \cos \frac{4\pi}{4} + \cos \frac{5\pi}{4} + \dots + \cos \frac{100\pi}{4}$$

$$S = 1 + \frac{\sqrt{2}}{2} + 0 - \frac{\sqrt{2}}{2} - 1 - \frac{\sqrt{2}}{2} + 0 + \frac{\sqrt{2}}{2} + 1 + \frac{\sqrt{2}}{2} + 0 - \frac{\sqrt{2}}{2} - 1 - \frac{\sqrt{2}}{2} + 0 + \frac{\sqrt{2}}{2} + \dots - 1$$

La somme des 8 premiers termes vaut 0, comme la somme des 8 suivants, etc. La somme des $8 \times 12 = 96$ premiers termes est donc nulle. Il reste la somme les 5 termes $\cos \frac{96\pi}{4} + \cos \frac{97\pi}{4} + \cos \frac{98\pi}{4} + \cos \frac{99\pi}{4} + \cos \frac{100\pi}{4}$ qui est égale à

$$1 + \frac{\sqrt{2}}{2} + 0 - \frac{\sqrt{2}}{2} - 1 = 0.$$

Script de Jules

```
S ← 0
Pour I allant de 0 à 100
    S ← S + cos(I*π/4)
FinPour
```

Script de Hana

```
S ← 1
Pour K allant de 1 à 100
    S ← S + cos(K*π/4)
FinPour
```

T.I. Basic

```
0→S
For(1,0,100)
S+cos(I*π/4)→S
End
Disp S
```

Basic Casio

```
0→S
For 0→I To 100
S+cos(I*π/4)→S
Next I
S
```

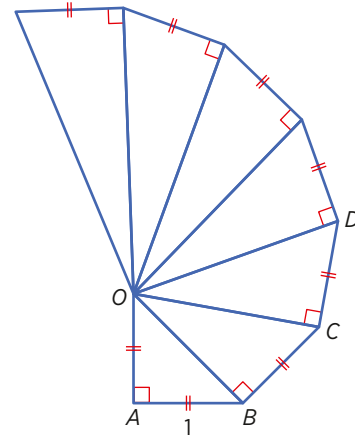


La spirale de Théodorus

Boucle bornée

La spirale de Théodorus est une figure qui permet de construire géométriquement les racines carrées des entiers consécutifs. On commence par construire un triangle rectangle et isocèle dont les côtés de l'angle droit mesurent 1 cm. L'hypoténuse de ce triangle constitue l'un des côtés de l'angle droit d'un nouveau triangle rectangle, comme indiqué sur la figure. L'autre côté de l'angle droit est à nouveau de longueur 1. On réitère le procédé en construisant à chaque fois un nouveau triangle rectangle dont un des côtés de l'angle droit est l'hypoténuse précédente, l'autre côté ayant pour longueur 1.

On veut déterminer au bout de combien de triangles la construction fait un tour complet.



PARTIE 1 Découverte du module Turtle de Python

Turtle est un module graphique du langage de programmation Python. Une tortue se déplace sur l'écran selon les instructions qui lui sont données dans le script. Les instructions utiles ici sont données ci-dessous :

Instruction	Fonction de cette instruction
reset()	Efface l'écran, recentre la tortue et remet les variables à 0
down()	Abaisse le crayon pour commencer à dessiner
up()	Relève le crayon pour pouvoir avancer sans dessiner
goto(x,y)	Se positionne au point(x,y)
forward(distance)	Avance d'une distance donnée (en pixels)
backward(distance)	Reculé d'une distance donnée (en pixels)
left(angle)	Pivote vers la gauche de l'angle donné (en radians)
right(angle)	Pivote vers la droite de l'angle donné (en radians)

1 À l'aide des instructions ci-dessus, donner le rôle du script suivant.

```
import math # importe les éléments du module math (dont sqrt, cos et sin)
import turtle # importe tous les éléments du module turtle #
crayon=turtle.Turtle()
crayon.reset()
crayon.right(90)
crayon.forward(50)
crayon.left(90)
crayon.forward(50)
crayon.left(135)
h=math.sqrt(50*50+50*50)
crayon.forward(h)
```

La tortue descend de 50 pixels, fait un angle droit, parcourt à nouveau 50 pixels et rejoint la position du point initial. Ce script permet de construire le premier triangle de la spirale de Théodorus.

2 Implanter ce script sur un ordinateur et vérifier la réponse précédente.

3 Quelles sont les coordonnées (en pixels) des sommets de la figure construite ?

Origine (0, 0) puis A(0 ; -50) puis B(50 ; -50).

4 Calculer OC (en pixels).

Théorème de Pythagore dans OBC : $OC = \sqrt{50 * 50 + 50 * 50 + 50 * 50} = 50\sqrt{3}$.

5 Compléter le script précédent pour obtenir le triangle OBC .

crayon.right(180)

crayon.forward(h)

crayon.left(90)

crayon.forward(50)

crayon.goto(0,)

PARTIE 2 Construction de n triangles composant la spirale

On considère le script ci-contre :

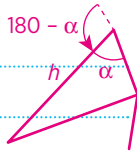
6 a) Implanter ce script puis l'exécuter avec $n = 2$, $n = 3$ puis $n = 4$.

b) Quel est le rôle des instructions ?

```
angle=180/math.pi*math.atan(h/50)
h=math.sqrt(h*h+50*50)
```

On pourra s'aider éventuellement d'un schéma.

La première calcule la mesure de l'angle α . La seconde calcule la longueur h de l'hypoténuse du nouveau triangle en construction.



```
import math # importe les éléments du module math (dont sqrt, cos et sin)
import turtle # importe tous les éléments du module turtle #
crayon=turtle.Turtle()
crayon.reset()
n=int(input("Donner le nombre de triangles désirés="))
crayon.right(90)
crayon.forward(50)
crayon.left(90)
crayon.forward(50)
crayon.left(135)
h=math.sqrt(50*50+50*50)
crayon.forward(h)
for i in range(1,n,1):
    crayon.right(180)
    crayon.forward(h)
    crayon.left(90)
    crayon.forward(50)
    angle=180/math.pi*math.atan(h/50)
    h=math.sqrt(h*h+50*50)
    crayon.left(180-angle)
    crayon.forward(h)
```

c) À l'aide de votre ordinateur, déterminer le nombre de triangles nécessaires pour faire « un tour complet ». 17 triangles.

PARTIE 3 Utilisation d'un tableur

7 Déterminer les mesures d'angles, en degrés, éventuellement arrondies au millième des angles :

a) \widehat{AOB} . OAB isocèle et rectangle en A donc $\widehat{AOB} = \widehat{ABO} = 45$ (degrés).

b) \widehat{BOC} . Dans BOC rectangle en B , $\sin(\widehat{BOC}) = \frac{1}{\sqrt{3}}$ donc $\widehat{BOC} = \sin^{-1}\left(\frac{1}{\sqrt{3}}\right) \approx 35,264$.

c) \widehat{COD} . Dans COD rectangle en C , $\sin(\widehat{COD}) = \frac{1}{4}$ donc $\widehat{COD} = \sin^{-1}\left(\frac{1}{4}\right) = 30$.

8 À l'étape n , on considère le triangle ci-contre :

On admet que la distance $OM = \sqrt{n}$ et $OC = \sqrt{n+1}$.



a) Justifier que la mesure de l'angle \widehat{MOC} est égale à $\sin^{-1}\left(\frac{1}{\sqrt{n+1}}\right)$. $\sin(\widehat{MOC}) = \frac{CM}{OC} = \frac{1}{\sqrt{n+1}}$

b) En créant une feuille de tableur à l'aide des indications ci-contre, retrouver le nombre d'étapes à partir duquel la construction a fait « un tour complet ». On vérifiera que ce nombre correspond à la valeur déterminée en Partie 2.

Colonne 1 : nombre de triangles
Colonne 2 : angle au centre
Colonne 3 : angles au centre cumulés

L'usage des connecteurs « ET » et « OU »

	ET	OU
Dans le langage usuel	Conjonction de coordination qui peut exprimer l'arrivée d'une nouvelle information, indiquer une énumération.	Conjonction de coordination qui exprime un choix, une alternative (« ou bien »), qui peut permettre la reformulation (« ou encore »). Exemples : – Dans un menu, on précise « fromage ou dessert ». OU a ici un sens exclusif , on ne peut pas profiter à la fois du fromage et du dessert. – Un serveur vous demande « Voulez-vous du lait ou du sucre avec votre café ? » OU a ici un sens inclusif . Vous pouvez vouloir l'un, l'autre ou les deux.
D'un point de vue logique Soient P et Q deux propositions ; chacune peut être soit Vraie, soit Fausse	La proposition P et Q est vraie si et seulement si les propositions P et Q sont toutes les deux vraies . Exemple : 2 est un nombre pair et inférieur à 3 est Vraie.	La proposition P ou Q est vraie si et seulement si l'une au moins des propositions P et Q est vraie . P ou Q est fausse si P et Q sont toutes les deux fausses. Exemple : On lance un dé classique : P : « La face obtenue est paire ». Q : « La face obtenue est multiple de 3 ». « P ou Q » est vraie pour les résultats 2, 3, 4, 6. OU a ici toujours un sens inclusif .
D'un point de vue algorithmique On considère deux conditions <cond1> et <cond2> ; chacune peut être soit Vérifiée, soit non Vérifiée	La condition (<cond1> et <cond2>) est vérifiée si et seulement si les conditions 1 et 2 sont toutes les deux vérifiées . Exemple : la condition ($X \geq 0$ et $X < 2$) est vérifiée si et seulement si $X \in [0 ; 2[$.	La condition (<cond1> ou <cond2>) est vérifiée si et seulement si l'une au moins des conditions 1 et 2 est vérifiée . Exemple : la condition $X \geq 0$ ou $X < 2$ est vérifiée pour tout nombre réel X. « Ou » a ici (comme en logique) un sens inclusif .

Exercice 1 Retrouver des effectifs

Variables et instructions élémentaires

Dans une petite ville, une série de sondages a été effectuée. Les séries de relevés statistiques ont été ensuite représentées graphiquement. Par malchance, les données numériques ont été perdues, seuls les diagrammes restent accessibles. La secrétaire de mairie souhaite retrouver les données chiffrées à partir des diagrammes.

1 On considère le diagramme circulaire ci-contre. La secrétaire a reporté les mesures d'angles au centre des secteurs. Comment peut-elle retrouver les fréquences, en %, correspondant à chaque valeur ? Compléter le tableau des fréquences.

On calcule la fréquence en % en divisant la valeur de l'angle au centre du secteur par la somme des angles (360°) et en multipliant par 100.

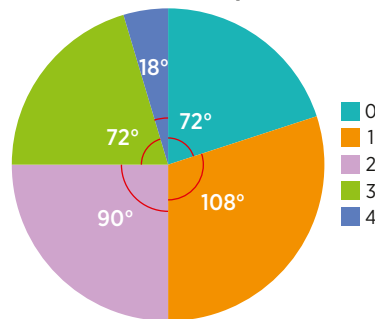
Par exemple, pour retrouver la fréquence des familles n'ayant pas d'enfants : $f = \frac{72}{360} \times 100 = 20$.

2 Compléter l'algorithme suivant afin qu'il calcule et affiche la fréquence correspondant à la mesure d'angle X saisie, en degrés.

3 Juliette a écrit un script avec le logiciel Scratch censé répondre au problème précédent. Ce script est-il correct ? Justifier votre réponse.

$\frac{100}{360} = \frac{5 \times 20}{18 \times 20} = \frac{5}{18}$. Les variables X et F sont bien les mêmes que dans l'algorithme de la question 2. le script de Juliette est correct.

Nombre d'enfants par famille



Nombre d'enfants par famille	0	1	2	3	4
Fréquence en %	20	30	25	20	5

Langage naturel

Variables X, F nombres flottants

Entrée Saisir X .

Traitement Affecter à F la valeur $\frac{X}{360} \times 100$.

Sortie Afficher F .

Scratch

```

quand est cliqué
demander "Quelle est la mesure en degrés de l'angle au centre ?" et attendre
mettre X à réponse
mettre F à X * 5 / 18
dire "La fréquence correspondante est" pendant 2 secondes
dire F
    
```

Exercice 2 Admis ou refusé ?

Instruction conditionnelle

Un examen comporte trois épreuves. Chaque candidat obtient une note A affectée du coefficient 2, une note B affectée du coefficient 3, puis une note C affectée du coefficient 5. Pour chaque épreuve, le nombre maximum de points est 20.

1 Compléter l'algorithme ci-contre de façon à ce qu'il calcule et affiche la moyenne M des 3 notes obtenues.

Langage naturel

Variables A, B, C nombres flottants

Entrées Saisir A, B, C

Traitement Affecter à M la valeur $\frac{2.A + 3.B + 5.C}{10}$.

Sortie Afficher M

2 Un candidat est admis à cet examen si et seulement si sa moyenne des trois épreuves est supérieure ou égale à 10. En ajoutant une variable R de type « chaîne de caractère » et une instruction conditionnelle « Si ... alors ... sinon », proposer un algorithme qui, une fois les notes A , B et C saisies, calcule la moyenne M et affiche le résultat du candidat sous la forme « Admis » ou « Refusé ».

Langage naturel

Variables : A, B, C, M nombres flottants, R chaîne de caractères
Entrées : Saisir A, B, C
Traitement : Affecter à M la valeur $\frac{2A + 3B + 5C}{10}$
 Si $M \geq 10$
 Affecter à R la valeur « Admis »
 Sinon
 Affecter à R la valeur « Refusé »
Sortie : Afficher R

3 En utilisant la fiche calculatrice, traduire l'algorithme précédent dans le langage de votre calculatrice. À l'aide de ce programme, déterminer si les candidats suivants sont reçus à cet examen.

	Note A	Note B	Note C	Moyenne	Résultat
Sam	8	9	11	9,8	Refusé
Hugo	12	8	11	10,3	Admis
Maxence	17	9	9	10,6	Admis

Exercice 3

Parité d'un nombre entier et application en statistiques

Instruction conditionnelle

1 On s'interroge sur le moyen de déterminer la parité d'un nombre N à l'aide d'un programme informatique. Le nombre N est saisi par l'utilisateur. Compléter les phrases suivantes.

Soit un entier naturel N , dire que N est pair, c'est dire que le quotient $N/2$ est un nombre entier.....

Soit un entier naturel N , dire que N est impair, c'est dire que le quotient $N/2$ n'est pas un nombre entier.....

2 Les programmes ci-contre permettent de déterminer si un entier saisi est pair ou impair. Entourer l'instruction conditionnelle correspondant à la détermination de la parité de l'entier N saisi. Implanter sur votre machine celui qui correspond à votre modèle et tester le programme sur quelques entiers.

T.I. Basic

```
PROGRAM:NOMBRE
:Input "SAISIR UN ENTIER N
":N
:If partEnt(N/2)=N/2
:Then
:Disp "CE NOMBRE EST PAIR"
:Else
:Disp "CE NOMBRE EST IMPAI
R"
```

Basic Casio

```
===== NOMBRE =====
"Saisir un entier N"?
->N
:If (Int (N÷2)=N÷2)
Then "Ce nombre est p
air"
Else "Ce nombre est i
IfEnde
```

3 On considère l'algorithme suivant.

Langage naturel

Entrée Saisir N
Traitement Si partie entière $\left(\frac{N}{2}\right) = \frac{N}{2}$
 Afficher « Le résultat est la moyenne des valeurs des rangs »
 Afficher $\frac{N}{2}$
 Afficher « et »
 Afficher $\frac{N}{2} + 1$
 Sinon
 Afficher « Le résultat est la valeur de rang : »
 Afficher $\frac{N+1}{2}$

Fin de l'algorithme

a) Exécuter mentalement cet algorithme pour $N = 24$.

L'algorithme affiche « Le résultat est la moyenne des valeurs des rangs $\frac{N}{2}$ et $\frac{N}{2} + 1$. »

b) Exécuter mentalement cet algorithme pour $N = 51$.

L'algorithme affiche « Le résultat est la valeur de rang $\frac{N+1}{2}$. »

c) En déduire le rôle de cet algorithme en précisant ce que représente le « résultat cherché » lorsque le nombre N est l'effectif total d'une série statistique.

Cet algorithme permet de déterminer la médiane d'une série statistique.

4 En utilisant la fiche calculatrice et les programmes donnés question 1, traduire ci-dessous cet algorithme puis implanter le programme correspondant sur la calculatrice. Vérifier ainsi les résultats obtenus en question 3.

T.I. Basic

```
Input "Saisir un entier N", N
If partEnt(N/2)=N/2
Then
Disp "Le résultat est la moyenne des valeurs
de rangs", N/2, "et", N/2+1
Else
Disp "Le résultat est la valeur de rang", (N+1)/2
End
```

Basic Casio

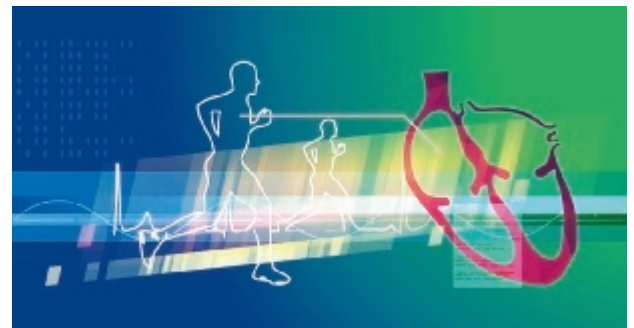
```
"Saisir un entier N"?→N
If (Int(N÷2)=N÷2)
Then "Le résultat est la moyenne des valeurs
de rangs":N÷2
"et":N÷2+1
Else "Le résultat est la valeur de rang": (N+1)÷2
IfEnd
```

Exercice 4 Fréquences cardiaques

Boucle bornée

Avant de proposer un cross de 5 km à ses élèves, un professeur d'E.P.S. décide de relever la fréquence cardiaque (en bpm : battements par minute) de chacun. Pour ce faire, par souci de gain de temps, il note le nombre de battements pendant une durée de 15 secondes des 10 premiers élèves. Il obtient les résultats ci-après.

Le professeur de mathématiques, dans le cadre de travaux interdisciplinaires, s'associe à lui pour traiter les données recueillies.



Numéro dossard	n° 1	n° 2	n° 3	n° 4	n° 5	n° 6	n° 7	n° 8	n° 9	n° 10
Nombre de battements pendant 15 secondes	16	19	20	25	18	27	22	24	23	26

1 Dans un premier temps, le professeur de mathématiques souhaite écrire un algorithme qui, une fois le nombre de battements pendant 15 secondes saisi, affiche la fréquence cardiaque par minute (en bpm). En notant X le nombre de battements pendant 15 secondes et F la fréquence cardiaque en bpm, proposer un algorithme qui réponde au problème.

Langage naturel

```
Variables X, F nombres flottants
Entrée(s) Saisir X
Traitement Affecter à F la valeur 4X
Sortie Afficher F
```

2 Au final, 150 élèves participeront au cross. Le professeur de mathématiques sait qu'il pourra écrire un algorithme qui permettra, en fin de cross et après recueil des fréquences, d'afficher la fréquence cardiaque maximale en bpm.

Compléter l'algorithme suivant pour qu'il calcule et affiche la fréquence cardiaque maximale des élèves portant les 10 premiers dossards (tableau de l'énoncé), ainsi que le numéro de dossard correspondant.

Langage naturel

Variables X, F, A, M nombres flottants.....

Initialisation M prend la valeur 0 ..
 A prend la valeur 0 ..

Traitement Pour I allant de 1 à 10 ..
 Saisir X
 F prend la valeur $4X$..
 Si $F > M$..
 M prend la valeur F ..
 A prend la valeur I ..
 FinSi
 FinPour

Sortie Afficher « La fréquence maximale relevée en bpm est »
 Afficher M ..
 « Le numéro du dossard correspondant est »
 Afficher A ..

3 À l'aide de la fiche calculatrice, traduire cet algorithme dans le langage de programmation de votre calculatrice. L'implanter et donner les valeurs de F et A affichées en sortie. Vérifier que cela correspond bien aux valeurs indiquées dans le tableau des dix premiers relevés.

$F = 108$ (bpm) et

$A = 6$ (numéro dossard)

4 Comment modifier l'algorithme précédent pour qu'il affiche également la fréquence cardiaque (bpm) minimale G de la série des valeurs relevées des 10 premiers élèves et le numéro B du dossard correspondant ?

T.I. Basic

$0 \rightarrow M$
 $0 \rightarrow A$
 For(1,1,10).....
 Prompt X.....
 $4 * X \rightarrow F$
 If $F > M$
 Then.....
 $F \rightarrow M$
 $I \rightarrow A$
 End.....
 End.....
 Disp "Fréquence max :":M.....
 Disp "Numéro dossard :":A.....

Basic Casio

$0 \rightarrow M$
 $0 \rightarrow A$
 For 1 \rightarrow I To 10.....
 ? \rightarrow X.....
 $4 * X \rightarrow F$
 If $F > M$
 Then $F \rightarrow M$
 $I \rightarrow A$
 IfEnd.....
 Next.....
 "Fréquence max:":M.....
 "Numéro dossard:":A.....

Il faut ajouter deux nouvelles variables : B pour le numéro du dossard (initialisée à 0), et G (la fréquence minimale, initialisée à 500 par exemple). Ajouter une instruction conditionnelle dans la boucle Pour existante.

Si $F < G$
 G prend la valeur F
 B prend la valeur I
 FinSi.....

5 Comment calculer alors l'étendue de la série des 10 premières valeurs ?

L'étendue de la série vaut $F - G$.

6 Comment les professeurs pourront-ils s'y prendre pour étudier les fréquences cardiaques des 150 élèves au moment du cross ? Proposer la modification d'une ligne de l'algorithme précédent.

Il faut modifier la première ligne de la boucle Pour : « Pour A allant de 1 à 150 ».

Exercice 5 L'atelier d'écriture de la MJC

Chaînes de caractères et listes (logiciel Python)

Une Maison des Jeunes et de la Culture (M.J.C.) propose des ateliers d'écriture. Le groupe de participants est composé de 8 personnes. Lors de la première séance, on demande à chaque participant de proposer un mot de la langue française qui exprime le mieux sa personnalité. Il peut s'agir d'un nom, d'un adjectif, ou encore d'un adverbe. Des badges seront ensuite réalisés pour chaque individu, sur lesquels seront inscrits les « mots » choisis lors de cette première séance. Leila propose le mot « Liberté » et Jérôme le mot « Enthousiasme ».

1 a) Implanter le script ci-contre en langage Python sur votre ordinateur.

```
texte=input("Entrez votre texte ici:")
N=len(texte)
print(N)
```

b) Exécuter ce programme avec le texte : Liberté. La valeur de N correspondante est : $N = 7$.

c) Exécuter ce programme avec le texte : Enthousiasme. La valeur de N correspondante est : $N = 12$.

d) Quel semble être le rôle de l'instruction `len(texte)` ?

Cette instruction calcule la longueur d'une chaîne de caractères.

2 La responsable de l'atelier d'écriture souhaite faire afficher les longueurs des mots pour les utiliser par la suite. Pour cela, elle crée une liste des 8 mots, à l'aide du script suivant.

```
liste1=["liberte","enthousiasme","futur","melancolie","chance","confiance","reve","ambition"]
```

Pour désigner la variable à la 3^e position dans cette liste, on note : `liste1[3]`

a) Compléter le script suivant pour qu'il affiche la liste 2 constituée des longueurs des mots de la liste 1 :

```
liste1=["liberte","enthousiasme","futur","melancolie","chance","confiance","reve","ambition"]
liste2=["","","","","","","",""]

for I in range(0,8): # attention, le premier mot est le mot de rang 0, le deuxième est de rang 1 ...etc#
    N=len(liste1[I]) # mais avec Python, on ajoute 1 à la dernière valeur attendue de I"
    liste2[I]= N
print (liste2)
```

b) Exécuter ce programme et noter ici la liste 2 affichée : `[7 12 5 10 6 9 4 8]`

3 Proposer un script qui calcule et affiche la longueur moyenne M de ces chaînes de caractères. On a $M = 7,625$.

```
liste1=["liberte","enthousiasme","futur","melancolie","chance","confiance","reve","ambition"]
```

```
liste2=["","","","","","","",""]
```

```
S=0
```

```
for I in range(0,8):
```

```
    N=len(liste1[I])
```

```
    S=S+N
```

```
    liste2[I]=N
```

```
print(S/8)
```

4 On considère maintenant la liste : `serie=[7,12,5,10,6,9,4,8]` qui contient 8 éléments. On considère l'algorithme ci-dessous.

Langage naturel

Variables	serie est une liste de nombres réels <i>i, j</i> sont des entiers naturels
Initialisation	serie prend [7,12,5,10,6,9,4,8]
Traitement	Pour <i>i</i> allant de 1 à 7 Affecter à <i>j</i> la valeur 8 Tant que <i>i</i> < <i>j</i> Si <code>serie[j]<serie[j-1]</code> alors Échanger les valeurs de <code>serie[j]</code> et de <code>serie[j-1]</code> FinSi Affecter à <i>j</i> la valeur <i>j</i> - 1 FinTantQue FinPour
Sortie	Afficher serie

a) Exécuter cet algorithme en complétant le tableau suivant. Pour *i* = 1,

	<i>j</i> = 8	<i>j</i> = 7	<i>j</i> = 6	<i>j</i> = 5	<i>j</i> = 4	<i>j</i> = 3	<i>j</i> = 2
Test	serie[8] <serie[7] Faux	serie[7] <serie[6] Vrai	serie[6] <serie[5] Vrai	serie[5] <serie[4] Vrai	serie[4] <serie[3] Vrai	serie[3] <serie[2] Vrai	serie[2] <serie[1] Vrai
Permutation	Non	Oui serie[6] et serie[7] permutent	Oui serie[5] et serie[6] permutent	Oui serie[4] et serie[5] permutent	Oui serie[3] et serie[4] permutent	Oui serie[2] et serie[3] permutent	Oui serie[1] et serie[2] permutent
serie	[7,12,5,10, 6,9,4,8]	[7,12,5,10, 6,4,9,8]	[7,12,5, 10,4,6,9,8]	[7,12,5,4, 10,6,9,8]	[7,12,4,5, 10,6,9,8]	[7,4,12,5, 10,6,9,8]	[4,7,12,5, 10,6,9,8]

b) Qu'a effectué l'algorithme pour *i* = 1 ?

Il a effectué un tri des valeurs 2 à 2 pour avoir la plus petite valeur en première position.

c) Recommencer pour *i* = 2. À la fin de l'exécution, on a : `serie=[4,5,12,5,10,6,9,8]`.

d) Quel est le rôle de cet algorithme ?

Cet algorithme permet de trier les valeurs d'une liste par ordre croissant.

e) Le script ci-après traduit l'algorithme précédent en langage Python. Implanter ce script sur un ordinateur et vérifier votre réponse à la question précédente. Entourer les instructions qui permettent l'échange de nombres dans la liste donnée.

```

serie=[7,12,5,10,6,9,4,8]
for i in range(0,8):
    j=7
    while i<j:
        if serie[j]<serie[j-1]:
            temp=serie[j]
            serie[j]=serie[j-1]
            serie[j-1]=temp
        j=j-1
print (serie)

```



Analyse fréquentielle et cryptanalyse

Boucles et instructions conditionnelles

Au IX^e siècle, le philosophe et scientifique arabe Al-Kindi (801-873) s'intéresse aux méthodes de cryptanalyse de données. Dans les archives ottomanes d'Istanbul, en 1987, on retrouve son « Manuscrit sur le déchiffrement des messages cryptographiques ».

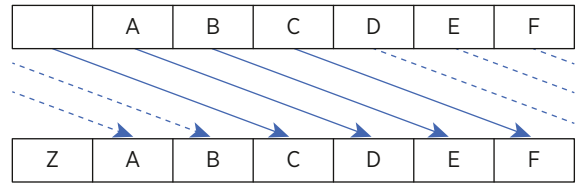
L'analyse fréquentielle est une méthode qui consiste à analyser les fréquences d'apparitions de symboles dans un texte crypté, pour les rapprocher ensuite des fréquences d'apparition des lettres dans la langue dans laquelle le message a été écrit à l'origine.

Le logiciel Python permet de s'intéresser aux longueurs des textes (les variables sont alors des chaînes de caractères), aux fréquences d'apparition de chaque lettre (ou symbole, selon les cas) dans ces textes. L'objectif de ce TP est de découvrir l'analyse fréquentielle, et de l'utiliser dans le but de décrypter un message. On ne s'intéresse ici qu'aux méthodes de chiffrement dites « de César ».

On choisit un nombre entier positif N . Chaque lettre du texte initial est alors remplacée par la lettre située N rangs « plus loin » dans l'alphabet.

Sur la figure ci-contre, on a choisi $N = 3$. N est appelé la « clé » du chiffrement.

Dans l'alphabet latin, on compte 26 clés possibles.



PARTIE 1

« n guugpvkgn guv kpxkukdng rqt ngu agwz. »

1 Le titre de cette partie est une phrase célèbre d'un écrivain et poète français. Il a été crypté par la méthode de chiffrement de César, mais la « clé » du code n'est pas donnée. Saurez-vous la retrouver ?

Pour faciliter la suite du TP, toutes les lettres sont inscrites en lettres minuscules, les symboles comme apostrophes ou parenthèses n'apparaissent pas.



BESOIN D'AIDE ?

On pourra commencer par s'interroger sur la première lettre de cette phrase : à quelle lettre pourrait correspondre ce « n » ?

Le « n » correspond à une lettre qui peut s'écrire seule, comme le « a », ou le « l » avec apostrophe.

Test avec le « l » : si on dit que « n » = « l », la clé est 2. Avec cette clé, la phrase décodée a un sens.

2 Une fois cette clé trouvée, écrire ci-dessous la phrase originale et préciser son auteur (grâce à une recherche internet si nécessaire).

Crypté	n	g	u	u	g	p	v	k	g	n	g	u	v	k	p	x	k	u	k	d	n	g	r	q	w	t	n	g	u	a	g	w	z
Original	l	e	s	s	e	n	t	i	e	l	e	s	t	i	n	v	i	s	i	b	l	e	p	o	u	r	l	e	s	y	e	u	x

Auteur : Antoine de Saint-Exupéry

3 Saisir le programme en langage Python ci-contre (les espaces entre les mots n'ont pas été saisis pour éviter qu'ils soient comptabilisés comme des « caractères »). À quoi correspond la valeur affichée ?

```
texte="nguugpvknguvkpxkukdng rqt ngu agwz"
len(texte)
print (len(texte))
```

La valeur affichée correspond à la longueur de la chaîne de caractères, soit 32.

4 Lorsque l'on veut décrypter un message codé par un chiffrement de César, on peut utiliser l'analyse fréquentielle. Quelle est la fréquence de la lettre « g » dans le texte codé « nguugpvknguvkpxkukdng rqt ngu agwz » ?

La fréquence de la lettre « g » est égale au nombre d'occurrences de « g » divisé par le nombre de caractères, soit $\frac{6}{32} = 0,19$.

5 Un script Python peut nous aider à calculer les fréquences d'apparition des différentes lettres utilisées. Pour ne pas fausser les calculs, les espaces présents dans le texte ont été supprimés.

a) Quel est le rôle du script ci-contre ?

Le script suivant permet de compter le nombre d'occurrences de « g » dans le texte codé.

```

texte="nguugpvkngnugvpxkukdngrqwtnguagwz"
N=0
for I in range(0,33):
    if texte[I]=="g":
        N=N+1
print (N)

```

b) Quelles lignes doit-on rajouter à la fin de ce script pour faire afficher la fréquence du caractère « g » dans le texte codé ?

On doit rajouter les lignes

$$F = \frac{N}{\text{len}(\text{texte})}$$

print(F)

c) Saisir ces nouvelles lignes, exécuter le script et vérifier que l'on retrouve la réponse donnée en 4.

6 D'après le site internet **dcode**, les fréquences d'apparition des lettres les plus utilisées dans les textes français sont les suivantes. Comparer la fréquence d'apparition de la lettre « g » dans la phrase précédente aux fréquences indiquées dans ce tableau.

Lettre	e	a	s	i	n
Fréquence	17,3 %	8,4 %	8,1 %	7,3 %	7,1 %

La fréquence calculée pour « g » est plus proche de la fréquence de la lettre « e » dans la langue française.

Remarque : cette méthode d'analyse fréquentielle est efficace, en général, sur des chaînes de caractères suffisamment longues pour que ces fréquences soient significatives, ce qui n'est pas tout à fait le cas dans notre exemple.

PARTIE 2 Cryptage du mot « avion » grâce au logiciel Python

On souhaite à présent crypter un message avec la méthode du chiffrement de César.

7 On commence par décider de la « clé » de cryptage : dans cette partie, la clé sera 4.
- Dans un script Python, on initialise trois chaînes de caractères :

```

alphabetini="abcdefghijklmnopqrstuvmxyz"
codage="EFGHIJKLMNOPQRSTUVWXYZABCD"
texte1="avion"

```

- On définit une fonction **permut** :

```

def permut(car):
    J=0
    while car!=alphabetini[J]: # note : "!=" signifie "différent"
        J=J+1
    return codage[J]

```

Quel est le rôle de cette fonction ?

Cette fonction permet de remplacer une lettre par son codage.

8 Voici le script restant à saisir :

```

for I in range(0,5): # attention, la première lettre est de rang 0#
    car=texte1[I]
    car2=permut(car)
    print (car2)

```

Expliquer le rôle de ce script.

Ce script permet de remplacer les lettres du mot avion par leur codage en conservant le même ordre des lettres.

9 Implanter ce programme sur un ordinateur. L'exécuter et vérifier que l'on obtient le cryptage du mot « avion » avec une clé de chiffrement de César égale à 4, c'est-à-dire : **EZMSR**.

Exercice 1 Programmer un instrument

Variables et instructions élémentaires



Bob, un artiste contemporain, décide de créer un programme qui reproduit les premières notes de l'Ode à la Joie de L. V. Beethoven. Avec le logiciel Scratch, il est possible de

faire jouer des notes de musique par 21 « instruments » différents. L'« instrument » n° 1 est le piano. Voici ci-contre le premier script qu'il réalise :

1 Saisir ce script sur un ordinateur et l'exécuter.

2 Bob souhaite faire jouer ces notes par un « instrument » choisi au hasard par le logiciel.

a) En créant une variable X qui correspond au numéro de l'instrument, modifier ce script de façon à ce que l'instrument soit choisi de manière aléatoire par le logiciel. (On notera « Jouer la note [...] » pour la partition complète.)

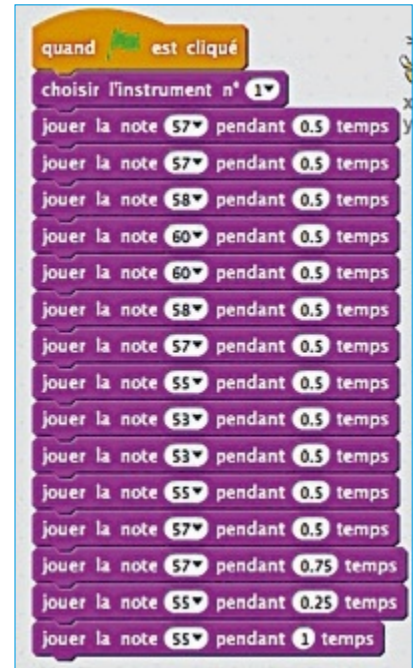
Scratch

Quand « Drapeau vert » est cliqué

Mettre X à nombre aléatoire entre 1 et 21

Choisir l'instrument n° X

« Jouer la note [...] »



b) Écrire le script qui permet d'obtenir un échantillon de 25 résultats obtenus par la répétition de façon indépendante de cette expérience (c'est-à-dire que cet extrait de morceau sera joué 25 fois de suite par des instruments choisis de manière aléatoire). Vérifier que le script réponde au problème.

Scratch

Quand « Drapeau vert » est cliqué

Répéter 25 fois

Mettre X à nombre aléatoire entre 1 et 21

Choisir l'instrument n° X

« Jouer la note [...] »

3 Parmi les « instruments » proposés par Scratch, ceux numérotés de 9 à 14 sont des instruments dits « à vent ».

a) Quelle est la probabilité p qu'un instrument choisi au hasard par le logiciel soit un instrument « à vent » ?

$$p = \frac{6}{21} = \frac{2}{7}$$

b) Modifier le script précédent (en créant une nouvelle variable S que l'on initialisera à 0) de façon à ce que soit calculée puis affichée la fréquence des résultats de l'expérience aléatoire impliquant un instrument « à vent » à chaque résultat.

Scratch

Quand « Drapeau vert » est cliqué

Mettre S à 0
Répéter 25 fois
Mettre X à nombre aléatoire entre 1 et 21
Si $8 < X$ et $X < 15$ alors
Ajouter à S 1
Choisir l'instrument n° X
« Jouer la note [...] »

c) La fréquence des instruments à vent choisi par le logiciel dans cet échantillon de 25 résultats peut-elle être égale à p ? Justifier la réponse.

f égale p si et seulement si le nombre N d'instruments à vent choisi sur les 25 instruments vérifie : $\frac{N}{25} = \frac{2}{7}$ ou encore $N = \frac{50}{7}$ or ce nombre rationnel n'est pas entier. La réponse est donc : non.

Exercice 2 Intervalle de fluctuation

Instruction conditionnelle et boucle bornée

Théo souhaite écrire un programme qui, une fois la proportion p et la taille n de l'échantillon donnés, vérifie les conditions « $n \geq 25$ et $0,2 \leq p \leq 0,8$ » et calcule puis affiche les bornes $a = p - \frac{1}{\sqrt{n}}$ et $b = p + \frac{1}{\sqrt{n}}$ de l'intervalle de fluctuation au seuil de 95 % pour les échantillons de taille n .

1 Combien de variables doit-on *a priori* créer pour l'écriture de cet algorithme ?

On peut en créer 4 : taille de l'échantillon : N ; probabilité : P ; borne a : A ; et borne b : B .

2 Théo peut-il n'utiliser qu'une instruction conditionnelle ?

Oui, « Si $N \geq 25$ et $0,2 \leq P \leq 0,8$ ».

3 Proposer un algorithme répondant au problème.

.... RAPPEL

On rappelle ici que pour un caractère dont la proportion dans une population donnée est p , pour $n \geq 25$ et $0,2 \leq p \leq 0,8$, la fréquence du caractère dans les échantillons de taille n appartient à l'intervalle $\left[p - \frac{1}{\sqrt{n}} ; p + \frac{1}{\sqrt{n}} \right]$ dans 95 % des cas.

Cet intervalle est appelé intervalle de fluctuation au seuil de 95 %.

Langage naturel

Variables N nombre entier
 P , A et B nombres flottants
Entrées Saisir N
Saisir P
Traitement Si $N \geq 25$ et $0,2 \leq P$ et $P \leq 0,8$
Affecter à A la valeur $P - \frac{1}{\sqrt{N}}$
Affecter à B la valeur $P + \frac{1}{\sqrt{N}}$
Afficher « Borne inférieure intervalle », A
Afficher « Borne supérieure intervalle », B
Sinon
Afficher « Les conditions ne sont pas vérifiées. »
FinSi

Fin de l'algorithme

- 4 Traduire cet algorithme et implanter le programme correspondant dans votre calculatrice.

T.I. Basic

```

Input "N=" , N
Input "PROBABILITE=" , P
If N ≥ 25 et 0,2 ≤ P et P ≤ 0,8
Then
P - 1/√(N) → A
P + 1/√(N) → B
Disp "BORNE INFERIEURE:" , A
Disp "BORNE SUPERIEURE:" , B
Else
Disp "CONDITIONS NON VERIFIEES"
End

```

Basic Casio

```

"N=" ↓
? → N ↓
"Probabilite P=" ↓
? → P ↓
If (N ≥ 25) And (0,2 ≤ P) (P ≤ 0,8) ↓
Then P - 1/√N → A ↓
P + 1/√N → B ↓
"Borne inferieure:" ↓
A ↓
"Borne superieure:" ↓
B ↓
Else "Conditions non verifiees" ↓
IfEnd

```

- 5 Qu'affiche la calculatrice à l'exécution du programme :

- a) pour $n = 20$ et $p = 0,4$? Elle affiche « Conditions non vérifiées ».
- b) pour $n = 100$ et $p = 0,35$? Elle affiche « Borne inférieure : 0,25 » et « Borne supérieure : 0,45 ».

Exercice 3

Épidémie de grippe

Boucle bornée

On estime que la grippe touche 20 % de la population française chaque année. Au 1^{er} janvier 2017, le nombre d'habitants en France est d'environ 67 millions.

- 1 Donner une estimation de nombre de personnes résidant en France touchées par la grippe en 2017.

Ce nombre est environ : $\frac{20}{100} \times 67\,000\,000 = 13\,400\,000$.

- 2 Déterminer l'intervalle de fluctuation de la fréquence des personnes touchées par la grippe sur des échantillons de taille 500.

$n = 500$ supérieur à 25, $p = 0,2$ alors $I = \left[0,2 - \frac{1}{\sqrt{500}} ; 0,2 + \frac{1}{\sqrt{500}} \right] \approx [0,155 ; 0,245]$.

- 3 On propose ici de faire une simulation de personnes touchées par la grippe sur un échantillon de 500 personnes.

- a) Expliquer le rôle du script ci-contre écrit en langage Python.

Ce script permet de simuler l'épidémie de grippe sur un échantillon de 500 personnes.

Python

```

import random
for N in range(1,501):
    X=random.randint(1,100)
    if X<=20:
        print ("personne atteinte par la grippe")
    else :
        print ("personne saine")

```

- b) Implanter puis exécuter ce script.

- c) Modifier le script pour faire afficher le nombre G puis la fréquence F des personnes malades dans l'échantillon de taille 500.

Python

```
import random
G=0
for l in range(1,501):
    X=random.randint(1,100)
    if X<=20:
        print("personne atteinte par la grippe")
        G=G+1
    else:
        print("personne saine")
print("Le nombre G de malades est")
print(G)
print("La fréquence F de malades sur l'échantillon de 500 personnes est")
print(G/500)
```

À l'exécution on obtient par exemple : $G = 113$ et $F = 0,226$.

- d) La fréquence F précédente appartient-elle à l'intervalle de fluctuation de fréquences au seuil de 95 % pour des échantillons de taille 500 ? Commenter le résultat.

La fréquence F observée sur l'échantillon appartient à l'intervalle I dans 95 % des cas.

Exercice 4 Roulette équilibrée ou non ?

Boucle non bornée

Une roulette de casino est constituée d'une bille, de 18 cases rouges, 18 cases noires et une case verte (le 0). Le croupier d'un casino souhaite vérifier si cette roulette est équilibrée.



- 1) Si l'on fait tourner la roulette 1 fois, en supposant qu'elle soit bien équilibrée, quelle est la probabilité que la bille se stabilise dans une case rouge ?

$$p = \frac{18}{37} \approx 0,486.$$

- 2) On considère des échantillons de N tirages.

- a) Déterminer l'intervalle de fluctuation de fréquence de la réalisation de l'évènement suivant R : « La bille arrive dans une case rouge » au seuil de 95 % en fonction du nombre de tirages N .

Si $N \geq 25$ avec $p = \frac{18}{37}$ alors l'intervalle est : $I = \left[\frac{18}{37} - \frac{1}{\sqrt{N}} ; \frac{18}{37} + \frac{1}{\sqrt{N}} \right]$ au seuil de 95 %.

- b) Sur 50 lancers, le croupier note que la bille s'est stabilisée 19 fois sur une case rouge. Peut-il remettre en question l'équilibrage de la roue ? Justifier soigneusement la réponse.

Pour $N = 50$ l'intervalle de fluctuation est : $I = \left[\frac{18}{37} - \frac{1}{\sqrt{50}} ; \frac{18}{37} + \frac{1}{\sqrt{50}} \right] \approx [0,345 ; 0,628]$ au seuil de 95 %.

Or ici, $f = \frac{19}{50} = 0,38$, $f \notin I$ donc il peut penser que la roue est bien équilibrée au seuil de 95 %.

- c) Déterminer l'amplitude de l'intervalle en fonction de N .

$$\text{L'amplitude de l'intervalle est } \Delta = \frac{18}{37} + \frac{1}{\sqrt{N}} - \left(\frac{18}{37} - \frac{1}{\sqrt{N}} \right) = \frac{2}{\sqrt{N}}$$

d) Le croupier souhaite être précis dans ses investigations. Il cherche à déterminer le plus petit entier N_0 (nombre de tirages minimum) pour lequel l'amplitude de l'intervalle de fluctuation est inférieure à 0,1. Une amie informaticienne lui propose de déterminer cet entier à l'aide d'un algorithme écrit en pseudo-langage, mais cet algorithme est incomplet. Compléter ce programme afin qu'il réponde au problème.

Pseudo-langage

```

N ← 1
Tant que  $\frac{2}{\sqrt{N}} \geq 0,1$ 
  N ← N + 1
FinTantQue
  
```

e) Traduire et implanter le programme correspondant à la calculatrice puis faire afficher la valeur de la variable N après l'exécution (on obtient ainsi le nombre cherché) N_0 .

T.I. Basic

1 → N

While 2/√(N) ≥ 0.1

N+1 → N

End

Disp. N

Basic Casio

1 → N ↓

While 2 ÷ √N ≥ 0.1 ↓

N+1 → N ↓

WhileEnd ↓

N ↓

On obtient à la fin de l'exécution : $N_0 = 401$.

f) Vérifier que l'amplitude de l'intervalle de fluctuation de fréquences de sortie de cases rouges sur des échantillons de taille N_0 est bien inférieure à 0,1 : Pour $N = 401$ et $p = \frac{18}{37}$, l'amplitude de l'intervalle est $\frac{2}{\sqrt{401}} \approx 0,09988$.

g) Retrouver le nombre N_0 en résolvant une inéquation dans l'ensemble des entiers naturels.

On résout l'inéquation $\frac{2}{\sqrt{N}} < 0,1$. La fonction inverse étant décroissante sur $]0; +\infty[$ ceci équivaut à : $\frac{\sqrt{N}}{2} > \frac{1}{0,1} \Leftrightarrow \sqrt{N} > 20$.

La fonction carré étant croissante sur $[0; +\infty[$ ceci équivaut à $N > 20^2$ soit $N > 400$.

Le plus petit entier N_0 répondant au problème est donc bien $N_0 = 401$.

Exercice 5

Égalité femmes/hommes ? Vous avez dit... égalité ?

Instruction conditionnelle et boucle bornée

À un concours de recrutement national pour un emploi administratif en 2017 se sont présentés 1 438 femmes et 704 hommes ayant tous et toutes le niveau requis.

500 personnes ont été admises, dont 188 hommes. La question est de savoir si le jury a respecté l'égalité des chances dans son mode de recrutement.

① Si l'égalité des chances est respectée, la probabilité d'admission d'une femme doit être égale à la probabilité d'admission d'un homme à compétences équivalentes. Montrer que la proportion d'admis au concours par rapport au nombre total de candidats est d'environ 0,233.

$$p = \frac{500}{1438 + 704} = \frac{500}{2142} \approx 0,233$$

② Déterminer l'intervalle de fluctuation I de fréquences d'admis sur un échantillon de 1 438 au seuil de 95 %. Peut-on penser que les femmes ont été avantagées ?

$$N = 1438 \text{ supérieur à } 25, \text{ on suppose } p \approx 0,233 \text{ alors } I = \left[0,233 - \frac{1}{\sqrt{1438}}; 0,233 + \frac{1}{\sqrt{1438}} \right] \approx [0,206; 0,26]$$

La fréquence des admis parmi les femmes est $f = \frac{500 - 188}{1438} \approx 0,217$. $f \in I$ donc a priori on peut penser que le mode

de recrutement des femmes correspond à la proportion globale pour les deux sexes, au seuil de 95 %.



3 Déterminer l'intervalle de fluctuation J de fréquences d'admis sur un échantillon de 704 au seuil de 95 %. Peut-on penser que les hommes ont été avantagés ?

$$N = 704 \text{ supérieur à } 25, \text{ on suppose } p \approx 0,233 \text{ alors } J = \left[0,233 - \frac{1}{\sqrt{704}} ; 0,233 + \frac{1}{\sqrt{704}} \right] \approx [0,195 ; 0,2706].$$

La fréquence des admis parmi les hommes est $f = \frac{188}{704} \approx 0,2670$. Cette fréquence appartient à J .

4 En utilisant les résultats précédents, donner un avis quant à la question du respect de l'égalité des chances suivant le genre dans le mode de recrutement du jury.

On peut penser que le mode de recrutement du jury a respecté l'égalité des chances suivant le genre, au seuil de 95 %.

5 Afin d'établir un document utile au jury pour les années à venir, on souhaite élaborer un algorithme qui, une fois les nombres de femmes candidates (A), hommes candidats (B), femmes reçues (C) et hommes reçus (D) donnés, teste la question du respect de la parité dans le mode de recrutement. Compléter l'algorithme suivant afin qu'il réponde au problème.

Langage naturel

Variables A, B, C et D nombres entiers

Entrées Saisir A, B, C et D

Traitement Affecter à E la valeur $A + B$

Affecter à G la valeur $C + D$

Affecter à P la valeur $\frac{C + D}{A + B}$

Affecter à Q la valeur $\frac{C}{A}$

Affecter à R la valeur $\frac{D}{B}$

Si $A \geq 25$ et $0,2 \leq P \leq 0,8$

Si $Q < P - \frac{1}{\sqrt{A}}$ ou $Q > P + \frac{1}{\sqrt{A}}$

Afficher « La fréquence des femmes admises paraît normale au seuil de 95 %... »

Sinon

Afficher « La fréquence des femmes admises paraît anormale au seuil de 95 %... »

FinSi

Si $B \geq 25$ et $0,2 \leq P \leq 0,8$

Si $R < P - \frac{1}{\sqrt{B}}$ ou $R > P + \frac{1}{\sqrt{B}}$

Afficher « La fréquence des hommes admis paraît normale au seuil de 95 %... »

Sinon

Afficher « La fréquence des hommes admis paraît anormale au seuil de 95 %... »

FinSi

Si $A < 25$ ou $B < 25$ ou $P < 0,2$ ou $P > 0,8$

Afficher « Les effectifs annuels ne permettent pas de se prononcer ».

Fin de l'algorithme



Comprendre le surbooking et l'éviter

Instructions conditionnelles et boucles

Technique apparue aux États-Unis en 1978, le surbooking est utilisé par plusieurs compagnies aériennes, en vue d'optimiser le remplissage des places à bord des avions et de réduire ainsi le tarif passager. Souvent, après avoir acheté leurs billets, de nombreux passagers ne se présentent pas à l'embarquement (pour des empêchements personnels), ce qui peut alors laisser quelques sièges vides. On estime, dans cette activité, que 80 % des clients ayant acheté leurs billets se présentent à l'embarquement. La compagnie aérienne se permet alors de vendre plus de billets que de places. La question est d'éviter que le nombre de présents soit supérieur au nombre de places dans l'avion.

PARTIE 1 Simulation d'embarquement

Pour un vol offrant 300 places assises, on suppose que 370 billets ont été vendus. On estime que la probabilité qu'un client qui a acheté un billet se présente effectivement à l'embarquement est de 0,8.

1 Compléter l'algorithme suivant de façon à ce qu'il simule le remplissage de l'avion le jour du départ.



Langage naturel

Variables	N, I nombres entiers, X nombre flottant
Initialisation	N prend la valeur 0
Début de l'algorithme	Pour I allant de 1 à 370..... Affecter à X un nombre aléatoire compris entre 0 et 1 Si $X \leq 0,8$ alors N prend la valeur $N+1$ FinSi
Sortie	Afficher N

2 a) Écrire puis implanter le programme correspondant à cet algorithme en langage Python.

Python

```
import random.....
N=0.....
for I in range(1,371):.....
    X=random.random().....
    if X<=0.8:.....
        N=N+1.....
print(N).....
```

b) Exécuter ce programme 5 fois et noter les nombres de passagers présents pour 300 places.

Exemples $N_1 = 299$ $N_2 = 302$ $N_3 = 295$ $N_4 = 298$ $N_5 = 303$

c) Entourer les nombres éventuels correspondant à des sureffectifs.

3 Proposer un programme en langage Python simulant 200 remplissages de cet avion de 300 places pour 370 places vendues. On fera compter le nombre de sureffectifs sur ces 200 remplissages.

4 Déterminer l'intervalle de fluctuation de fréquences des personnes effectivement présentes au seuil de 95 % pour des échantillons de taille 370.

$$N = 370 \text{ supérieur à } 25, p = 0,8 : \\ \text{on a } I = \left[0,8 - \frac{1}{\sqrt{370}} ; 0,8 + \frac{1}{\sqrt{370}} \right] \approx [0,748 ; 0,852].$$

Python

```
import random
S=0
for J in range(0,201):
    N=0
    for I in range(1,371):
        X=random.random()
        if X<=0.8:
            N=N+1
    print(N)
    if N>=300:
        S=S+1
print(S)
```

PARTIE 2

Intervalle de fluctuation de fréquences sur des échantillons de taille N , avec $N \geq 25$

5 La compagnie vend N places (avec N supérieur à 25). En supposant que la probabilité qu'une personne ayant acheté un billet se rende à l'embarquement soit égale à 0,8, déterminer l'intervalle de fluctuation de fréquence pour l'échantillon de taille N (les bornes de cet intervalle seront exprimées en fonction de N).

$$\left[0,8 - \frac{1}{\sqrt{N}} ; 0,8 + \frac{1}{\sqrt{N}} \right]$$

6 Dédurre du résultat précédent que le nombre maximum de présents à l'embarquement en fonction de N au seuil de 95 % est égal à $N \left(0,8 + \frac{1}{\sqrt{N}} \right)$.

Sur les N places vendues, la fréquence maximale (au seuil de 95 %) de présents est $f = 0,8 + \frac{1}{\sqrt{N}}$. Le nombre maximal de personnes présentes à l'embarquement est donc $N \times f = N \times \left(0,8 + \frac{1}{\sqrt{N}} \right)$.

7 Sachant que l'avion dispose de 300 places, compléter l'algorithme ci-dessous pour qu'il détermine le plus grand entier N_0 pour lequel $N \left(0,8 + \frac{1}{\sqrt{N}} \right) \leq 300$.

Langage naturel

Variable	N nombre entier
Initialisation	N prend la valeur 25
Traitement	Tant que $N \left(0,8 + \frac{1}{\sqrt{N}} \right) \leq 300$ N prend la valeur $N + 1$ FinTantQue
Sortie	Afficher N

8 Pour utiliser la fonction « racine carrée » sur Python, commencer par importer le module Math avec l'instruction suivante : `from math import *`

a) Écrire le script correspondant à cet algorithme en langage Python et l'exécuter.

b) À l'exécution, on obtient : $N = 352$.

Python

```
from math import *
N=25
while N*(0.8+1/sqrt(N))<=P:
    N=N+1
print(N)
```

9 Interpréter le résultat précédent en revenant au contexte de l'énoncé.

Si elle veut éviter de refuser des passagers (au seuil de 95 %), la compagnie doit vendre au maximum 352 billets (pour 300 places).

PARTIE 3 Généralisation pour un avion à P places

10 En vous aidant de la Partie 2, élaborer un programme en langage Python qui détermine, en fonction du nombre P de places disponibles dans l'avion (P saisi par l'utilisateur) le nombre maximal de places que la compagnie peut vendre pour éviter le refus d'embarquer des passagers au seuil de 95 %.

11 Exécuter l'algorithme précédent avec $P = 255$ (places). Donner alors le nombre maximal de tickets que la compagnie devrait vendre pour éviter le sureffectif (au seuil de 95 %). 298.

Python

```
from math import *
P=int(input("Nombre de places dans l'avion"))
N=25
while N*(0.8+1/sqrt(N))<=P:
    N=N+1
print(N)
```

PARTIE 4 Recherche du nombre de places disponibles dans un avion



Sachant que la compagnie, avec les mêmes hypothèses, peut se permettre de vendre un maximum de 410 billets pour un vol en évitant le sureffectif (au seuil de 95 %), retrouver le nombre de places disponibles dans l'avion.

On peut définir une fonction du paramètre P , et avec une boucle itérative avec sortie de boucle conditionnelle, faire chercher ce nombre de places comme dans le programme ci-dessous.

À l'exécution on obtient 348.

C'est donc pour un nombre de places disponibles égal à 348 que la compagnie peut vendre un maximum de 410 billets.

Python


```
from math import *
def nombremax(P):
    N=25
    while N*(0.8+1/sqrt(N))<=P:
        N=N+1
    return N
print(N)
P=25
while nombremax(P)!=410:
    P=P+1
print(P)
```

Exercice 1

Pile ou Face

Instruction conditionnelle et boucle bornée

On lance une pièce de 1 € bien équilibrée à Pile ou Face.
On considère le script ci-contre écrit avec le logiciel Scratch.

L'instruction  renvoie un nombre entier aléatoire égal à 0 ou 1.



- 1 Saisir ce script sur un ordinateur et l'exécuter.
- 2 Justifier que ce script permet de simuler le lancer de la pièce de 1 € bien équilibrée.

À chaque lancer, $p(\text{Pile}) = p(\text{Face}) = \frac{1}{2}$ et à chaque fois que le drapeau vert est pressé, $p(X=0) = p(X=1) = \frac{1}{2}$.

- 3 On s'intéresse maintenant à l'expérience aléatoire « trois lancers successifs à Pile ou Face d'une pièce bien équilibrée ».

a) Donner l'ensemble de toutes les issues possibles de cette expérience aléatoire.

$\Omega = \{PPP, PPF, PFP, PFF, FPP, FPF, FFP, FFF\}$.

b) Modifier le script précédent pour qu'il permette de simuler trois lancers consécutifs de la pièce de 1 €.

Scratch

Quand « Drapeau vert » est cliqué

Répéter 3 fois

Mettre X à nombre aléatoire entre 0 et 1

Si X = 0 alors

Dire Pile pendant 2 secondes

Sinon

Dire Face pendant 2 secondes

Attendre 2 secondes

c) On considère l'événement suivant A : « On a obtenu exactement une fois Pile. ». Déterminer la probabilité de l'événement A en justifiant votre réponse.

Sur les 8 issues, seules les issues PFF, FPF et FFP réalisent l'événement A. La pièce étant bien équilibrée, on est en situation d'équiprobabilité d'où $p(A) = \frac{\text{nombre de cas favorables}}{\text{nombre de cas possibles}} = \frac{3}{8} = 0,375$.

Exercice 2 Boucle non bornée

Un sac contient des jetons indiscernables au toucher. Parmi ces jetons, 4 sont blancs, les n autres sont noirs (avec n entier strictement positif). On tire au hasard un jeton de ce sac et l'on note sa couleur. On considère l'événement B : « Le jeton tiré est blanc ».

1 Exprimer le nombre total de jetons du sac en fonction du nombre n : le nombre total de jetons est $n + 4$.

2 Déterminer la probabilité de l'évènement B dans chacun des cas suivants.

• Pour $n = 1$: $p(B) = \frac{4}{4+1} = \frac{4}{5} = 0,8$ • Pour $n = 2$: $p(B) = \frac{4}{4+2} = \frac{4}{6} = \frac{2}{3}$ • Pour $n = 3$: $p(B) = \frac{4}{4+3} = \frac{4}{7}$

3 Compléter la phrase suivante : « Plus n est grand, plus $p(B)$ est petit (tout en restant positif ou nul) ».

4 Compléter l'algorithme ci-contre afin qu'il calcule et affiche la probabilité de l'évènement B pour toutes les valeurs de n comprises entre 1 et 100.

Langage naturel

Variables	N nombre entier P nombre flottant
Début de l'algorithme	Pour N allant de 1 à 100 Affecter à P la valeur $\frac{4}{N+4}$ Afficher P FinPour
Fin de l'algorithme	

5 On souhaite déterminer le plus petit entier n_0 tel que pour tout $n \geq n_0$ on a $p(B) < 0,01$.

a) Compléter l'algorithme ci-contre afin qu'il détermine la valeur de n_0 .

Langage naturel

Variable	N nombre entier
Initialisation	N prend la valeur 0 P prend la valeur 1
Traitement	Tant que $P \geq 0,01$ N prend la valeur $N+1$ P prend la valeur $\frac{4}{N+4}$ Fin TantQue
Sortie	Afficher N

b) Parmi les deux scripts ci-contre, lequel traduit l'algorithme précédent ?

c) Implanter le script répondant au problème. Quelle est la valeur de N affichée en sortie ? $N = 397$.

d) Interpréter ce résultat dans le contexte de l'énoncé.

C'est pour un nombre de jetons noirs supérieur ou égal à 397 que la probabilité de tirer un jeton blanc dans ce sac passe sous le seuil de 0,01, soit 1 %.

e) Vérifier le résultat par le calcul.

On peut résoudre l'inéquation $\frac{4}{N+4} < 0,01 \Leftrightarrow \frac{4}{N+4} - 0,01 < 0$.

En réduisant le membre de gauche à un seul quotient : $\frac{4}{N+4} - \frac{0,01(N+4)}{N+4} < 0$
 $\frac{3,96 - 0,01N}{N+4} < 0$

Ici, $N > 0$ donc $N+4 > 0$ et le signe du membre de gauche est le signe de $3,96 - 0,01N$.

Or, $3,96 - 0,01N < 0 \Leftrightarrow 0,01N > 3,96$. En multipliant les deux membres par 100, on retrouve $N > 396$.

Le plus petit entier n_0 solution est donc bien $n_0 = 397$.

Script 1 Python

```
N=0
P=1
while P>=0.01:
    N+1=N
    P=4/(N+4)
print(P)
```

Script 2 Python

```
N=0
P=1
while P>=0.01:
    N=N+1
    P=4/(N+4)
print(N)
```

Exercice 3

Le nombre mystère : programmation d'un petit jeu à la calculatrice

Instruction conditionnelle et boucle bornée

Pour ses débuts dans la programmation, Tom décide de programmer le jeu suivant : une machine va choisir au hasard un nombre entier X compris entre 0 et 100 (0 et 100 sont inclus). Le joueur doit deviner ce nombre en un minimum de coups. À chaque proposition, trois réponses de la machine sont possibles :

- **Réponse 1** : le nombre X est strictement supérieur à la proposition du joueur.
- **Réponse 2** : le nombre X est strictement inférieur à la proposition du joueur.
- **Réponse 3** : le nombre X est égal à la proposition du joueur (dans ce dernier cas, le jeu s'arrête).

On cherche à écrire un algorithme pour programmer ce jeu à la calculatrice.

- 1 En supposant que le joueur propose un nombre choisi au hasard entre 0 et 100, quelle est la probabilité qu'il propose le nombre X au premier coup ?

Sur les 101 issues, une seule est favorable donc $p = \frac{1}{101}$.

- 2 Si le nombre mystère est égal à 50, en supposant que le joueur donne une proposition au hasard, quelle est la probabilité que la réponse donnée soit la Réponse 2 ?

Nombres de cas favorables : 50 ; nombre de cas possibles : 101. On en déduit $p = \frac{50}{101}$.

- 3 a) Tom teste le concept de son jeu avec Kim et choisit un nombre. La première proposition de Kim est 40. Tom donne la Réponse 1. À quel intervalle appartient alors le nombre X ?

Il appartient à l'intervalle $[40 ; 100]$.

- b) Kim cherche une stratégie pour optimiser son nombre de coups. Quel nombre devrait-elle alors proposer lors du coup suivant ?

Le centre de l'intervalle $[40 ; 100]$, soit $\frac{40 + 100}{2} = 70$.

- 4 Tom commence à écrire une partie de l'algorithme pour programmer ce jeu.

- a) À quoi correspond la variable P ?

P correspond au nombre choisi par le joueur.

- b) Quel est le rôle de la variable N ?

N correspond au nombre de coups.

- c) Expliquer l'instruction $N \leftarrow N + 1$: À chaque proposition

du joueur, le nombre de coups est augmenté de 1.

- d) Quelle est la valeur de N avant la première proposition ? $N = 0$.

- e) À quels intervalles appartiennent les valeurs de P et de X ? $P \in [0 ; 100]$ et $X \in [0 ; 100]$.

- f) En utilisant l'algorithme ci-contre, compléter le programme (en T.I. Basic ou Basic Casio) correspondant à ce jeu du nombre mystère.

Pseudo-langage

```
Tant que  $P \neq X$ 
  Saisir « Entrer  $P$  »
  Si  $P < X$ 
    Afficher « Le nombre mystère est supérieur. »
  Sinon
    Si  $P > X$ 
      Afficher « Le nombre mystère est inférieur. »
  FinSi
   $N \leftarrow N + 1$ 
FinTantQue
```

T.I. Basic

```
PROGRAM:MYSTERE
:ent(101*NbrAléat)→X
:0→N
:-1→P
:While P≠X
:Input "SAISIR P",P
:If P<X
:Then
:Disp "LE NOMBRE MYSTERE E
ST SUPERIEUR"
:Else
:If P>X
:Disp "LE NOMBRE MYSTERE E
ST INFÉRIEUR"
:End
:N+1→N
:End
:Disp "C EST GAGNE EN",N,"
COUPS"
```

Basic Casio

```
===== MYSTERE =====
RanInt#(0,100)→X
0→N
-1→P
While P≠X
  "Saisir P"
  ?→P
  If P<X
  Then "LE NOMBRE MYSTE
RE EST SUPERIEUR"
  Else If P>X
  Then "LE NOMBRE MYSTE
RE EST INFÉRIEUR"
  IfEnde
  IfEnde
  N+1→N
WhileEnde
"C EST GAGNE EN"
N
```



BESOIN D'AIDE?

Pour obtenir un nombre pseudo-aléatoire compris entre 0 et 100 sur les calculatrices :

T.I. : `ent(101*NbrAléat)`

Casio : `RanInt#(0,100)`

Exercice 4 Tirage au sort

Instruction conditionnelle et boucle bornée



Une urne contient 100 boules indiscernables au toucher numérotées 00, 01, 02, 03, 04, ..., 10, 11, ..., 99. On tire une boule au hasard et on lit le numéro obtenu. On considère les événements suivants :

A : « Le chiffre 1 figure dans le numéro. »

B : « Le chiffre 2 figure dans le numéro. »

1 Déterminer la probabilité de l'événement A, puis celle de l'événement B.

$$p(A) = \frac{19}{100} \text{ et } p(B) = \frac{19}{100}$$

2 Décrire l'événement \bar{A} , contraire de l'événement A, puis en déterminer la probabilité.

$$\bar{A} : \text{« Le numéro ne comporte pas le chiffre 1. »} \quad p(\bar{A}) = 1 - p(A) = 1 - \frac{19}{100} = \frac{81}{100}$$

3 Décrire l'événement $A \cap B$ puis en déterminer la probabilité.

$$A \cap B : \text{« Le numéro est composé du chiffre 1 et du chiffre 2. »} \quad \text{Deux issues : 12 et 21. } p(A \cap B) = \frac{2}{100} = \frac{1}{50} = 0,02$$

4 Décrire l'événement $A \cup B$ puis en déterminer la probabilité.

$A \cup B$: « Le numéro comporte le chiffre 1 ou le chiffre 2. »

$$p(A \cup B) = p(A) + p(B) - p(A \cap B) = \frac{19}{100} + \frac{19}{100} - \frac{2}{100} = \frac{36}{100} = 0,36$$

5 Compléter l'algorithme ci-contre de façon à ce qu'il simule le tirage d'une boule au hasard et qu'il teste si l'événement $A \cap B$ est réalisé.

Langage naturel

Variables	X nombre entier
Traitement	Affecter à un entier aléatoire compris entre 0 et 99 Afficher X Si $X=12$ ou $X=21$ Afficher « A et B est réalisé. » Sinon afficher « A et B n'est pas réalisé. » FinSi

Fin de l'algorithme

6 Traduire l'algorithme ci-dessus dans le langage de programmation de votre calculatrice et vérifier qu'il fonctionne.



BESOIN D'AIDE?

En T.I. Basic on obtient un nombre entier aléatoire compris entre 0 et 99 avec : `ent(NbrAléat*100)`

En Basic Casio on obtient un nombre entier aléatoire compris entre 0 et 99 avec : `RanInt#(0,99)`

T.I. Basic

```
Ent(NbrAléat*100)→X
Disp X
If X=12 ou X=21
Then
Disp "A ET B EST REALISE"
Else
Disp "A ET B N'EST PAS REALISE"
End
```

Basic Casio

```
RanInt#(0,99)→X
X↓
If (X=12) Or (X=21)↓
Then "A ET B EST REALISE"↓
Else "A ET B N'EST PAS REALISE"↓
IfEnd
```

Exercice 5

Coordonnées de points

Instruction conditionnelle

Dans le plan repéré, on choisit un point au hasard dont les coordonnées sont des nombres entiers aléatoires compris entre 0 et 3. On s'intéresse à la probabilité de l'évènement E : « Le point appartient à la droite (d) d'équation $y = x + 1$. ».

1 Dénombrer les couples de coordonnées possibles (on pourra s'aider d'un arbre).

Pour l'abscisse : 4 possibilités, pour l'ordonnée : 4 possibilités soit un nombre total de $4 \times 4 = 16$ issues possibles.

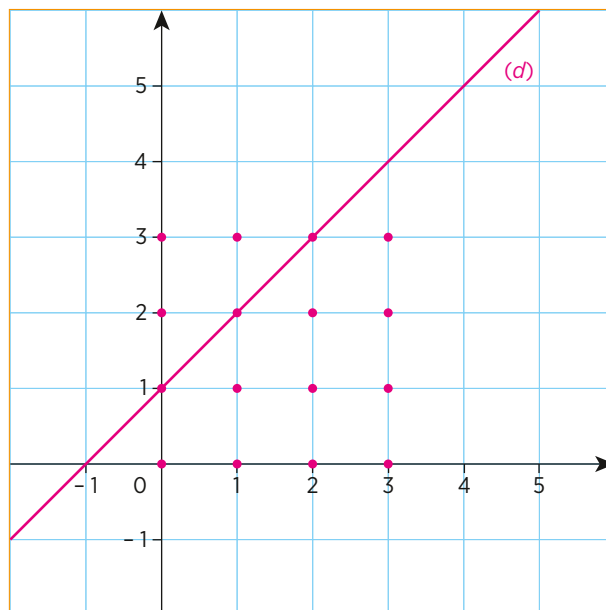
2 Le point $A(1 ; 3)$ appartient-il à la droite (d) ? Justifier votre réponse.

$x_A = 1$ et donc $x_A + 1 = 1 + 1 = 2$. On a $y_A \neq x_A + 1$; le point A n'appartient pas à la droite (d) .

3 Dans le repère ci-contre, placer en tous les points correspondants aux issues possibles de cette expérience et tracer en rouge la droite (d) .

4 Cette situation est-elle une situation d'équiprobabilité ? Justifier votre réponse.

Oui, les seize issues sont équiprobables, car les coordonnées sont choisies de manière aléatoire entre 0 et 3.



5 Quelle est la probabilité de l'évènement E ? Justifier la réponse.

Trois des seize points appartiennent à la droite (d) donc $p(E) = \frac{3}{16}$.

6 Le module *random* de Python permet d'utiliser des fonctions générant des nombres aléatoires et la bibliothèque *matplotlib* permet de placer des points et de tracer des courbes. Quel est le rôle du programme ci-contre ?

Python

```
import random
import matplotlib.pyplot as mpy
abscisse=random.randint(0,3)
ordonnee=random.randint(0,3)
mpy.plot(abscisse,ordonnee,'ro')
```

Ce script permet de placer dans un repère un point dont les coordonnées sont des entiers aléatoires compris entre 0 et 3.

7 Implanter ce programme sur un ordinateur puis le compléter de façon à ce qu'il détermine si le point dont les coordonnées sont choisies au hasard appartient à la droite (d) .

À la suite de ce programme, on ajoute :

```
if ordonnee==abscisse+1:
    print(« Le point appartient à la droite (d) »)
else:
    print(« Le point n'appartient pas à la droite (d) »)
```

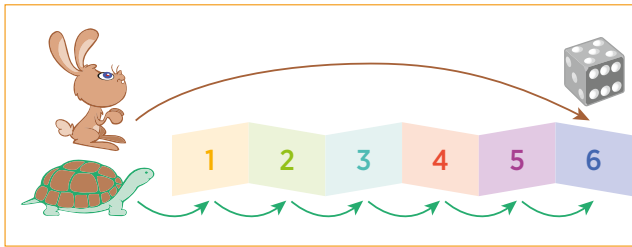
Python

```
import random
import matplotlib.pyplot as mpy
abscisse=random.randint(0,3)
ordonnee=random.randint(0,3)
mpy.plot(abscisse,ordonnee,'ro')
```



Rien ne sert de courir ?

Boucle non bornée



Règle du jeu (2 joueurs)

L'un des joueurs représente le lièvre, l'autre la tortue.

À chaque tour, on lance un seul dé bien équilibré :

- si le 6 sort, le lièvre avance et gagne immédiatement la partie ;
- sinon, la tortue avance d'une case.

Pour gagner, la tortue doit avancer 6 fois consécutives d'une case.

Pour avoir le plus de chance de gagner, préféreriez-vous représenter le lièvre ou la tortue à ce jeu ?

PARTIE 1 Simuler une partie du jeu

- 1 À chaque lancer, quelle est la probabilité
- de l'événement T : « la tortue avance d'une case » ? $p(T) = \frac{5}{6}$
 - de l'événement L : « le lièvre sort et gagne » ? $p(L) = \frac{1}{6}$

2 a) En utilisant le petit script ci-contre, simuler cinq parties de ce jeu en binôme. Noter le nombre de parties gagnées par la tortue : 2

b) Est-il pertinent à ce stade d'établir une conjecture en désignant l'animal qui semble être avantagé à ce jeu ?

Non, puisque la taille de l'échantillon est insuffisante (N=5).

3 a) Regrouper ci-dessous les résultats de tous les binômes.

Animal	Lièvre	Tortue	Total
Nombre de parties gagnées	55	35	5 × 18

Python

```
import random
D=random.randint(1,6)
print(D)
```

b) Conjecturer quel animal semble avantagé.

A priori le lièvre semble avantagé.

4 L'algorithme suivant permet de simuler une partie du jeu.

Langage naturel

Variables D et C nombres entiers

Initialisation D prend la valeur 0
C prend la valeur 0

Traitement Tant que D < 6 et C < 6
Affecter à D un nombre aléatoire compris entre 1 et 6
Affecter à C la valeur C + 1
Afficher D
FinTantQue
Si C = 6 faire
Afficher « La tortue a gagné. »
Sinon afficher « Le lièvre a gagné. »

Fin de l'algorithme

Python

```
import random
D=0
C=0
while D<6 and C<6:
    D=random.randint(1,6)
    C=C+1
    print(D)
if C==6:
    print("La tortue a gagné.")
else:
    print("Le lièvre a gagné.")
```

a) Traduire cet algorithme en langage Python.

b) Quel rôle joue la variable D ? Elle stocke la valeur de la face du dé à chaque lancer.

c) Quel rôle joue la variable C ? Elle stocke la position de la tortue (numéro de la case).

PARTIE 2 Simulation de 100 parties, puis de 1 000 parties

5 En utilisant l'algorithme précédent et en créant deux nouvelles variables, I le compteur d'itérations et S le nombre de parties gagnées par le lièvre, compléter l'algorithme qui permet de simuler 100 parties de ce jeu.

6 Implanter le script correspondant et noter le nombre de parties gagnées par le lièvre sur 100 parties :

(Exemple) $S = 59$.

7 Modifier le programme précédent pour simuler 1 000 parties. En sortie on a :

(Exemple) $S = 567$.

8 Confirmez-vous la conjecture émise en Partie 1 question 4.b) ?

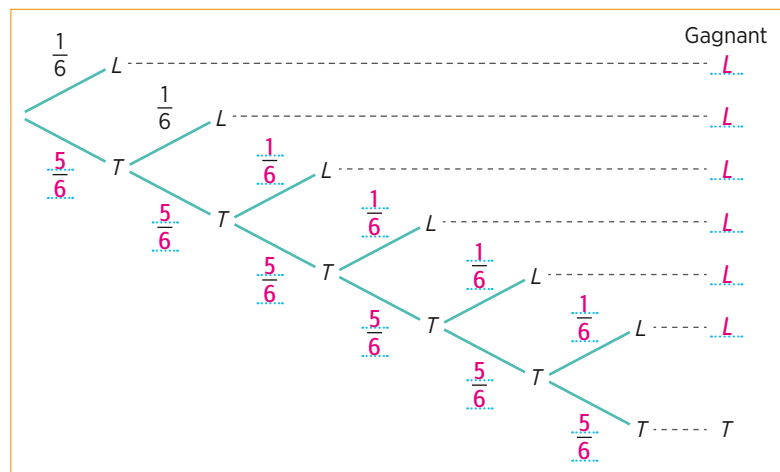
Oui, le lièvre semble avoir deux fois plus de chance de gagner que la tortue.

Langage naturel

Variables	D, C, S et I nombres entiers
Initialisation	S prend la valeur 0
Traitement	Pour I allant de 1 à 100 D prend la valeur 0 C prend la valeur 0 Tant que $D < 6$ et $C < 6$ Affecter à D un nombre aléatoire compris entre 1 et 6 Affecter à C la valeur $C + 1$ Afficher D FinTantQue Si $C = 6$ Afficher « La tortue a gagné » Sinon afficher « Le lièvre a gagné » Affecter à S la valeur $S + 1$. FinSi FinPour
Sortie	Afficher S

PARTIE 3 Calcul des probabilités des événements A : « La tortue gagne » et B : « Le lièvre gagne »

9 Compléter l'arbre pondéré suivant qui modélise une partie de jeu :



10 En admettant que la probabilité d'une issue est égale au produit des probabilités rencontrées sur le chemin menant à cette issue, déterminer $p(A)$, puis en déduire $p(B)$.

$$p(A) = \left(\frac{5}{6}\right)^6 \approx 0,33 \text{ et donc } p(B) = 1 - \left(\frac{5}{6}\right)^6 \approx 0,67.$$

11 Les conjectures émises dans les parties précédentes sont-elles démontrées ?

Oui, les résultats précédents prouvent que le lièvre a environ deux fois plus de chance de gagner à ce jeu que la tortue.

Si ... alors ... : implication et instruction conditionnelle

1 Point de vue logique

■ Implication

La proposition « **Si P alors Q** » est appelée **implication**. Cette proposition est vraie **si P implique Q**. On note $P \Rightarrow Q$.

Exemple : « Si les diagonales d'un quadrilatère ont même milieu alors ce quadrilatère est un parallélogramme. »

Cette proposition est **vraie**.

■ Réciproque

L'implication $P \Rightarrow Q$ a pour **réciproque** $P \Leftarrow Q$.

Exemple : La réciproque de notre proposition précédente est :

« **Si un quadrilatère est un parallélogramme, alors ses diagonales se coupent en leur milieu.** »

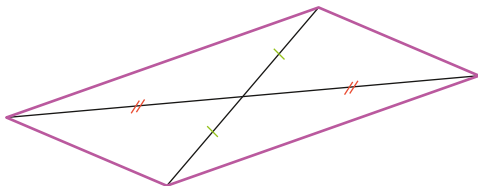
Cette réciproque est **vraie**.

Attention : la **réciproque** d'une proposition vraie **n'est pas toujours vraie**.

■ Équivalence

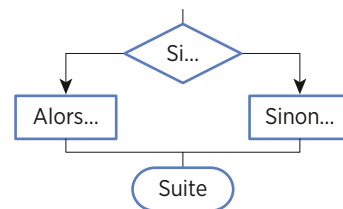
Si $(P \Rightarrow Q)$ et $(Q \Rightarrow P)$ alors on note $P \Leftrightarrow Q$.
P et Q sont dites **équivalentes**.

Exemple : Dire que « les diagonales d'un quadrilatère ABCD ont même milieu » **équivalent à dire** que « les diagonales [AC] et [BD] ont même milieu ».



2 Point de vue algorithmique

L'**instruction conditionnelle** « **Si <condition> alors <traitement>** » est une fonction d'un langage de programmation qui effectue différentes actions en fonction de l'évaluation d'une condition.



Un **booléen** est un type de variable à deux états : **VRAI** ou **FAUX**. Certains langages utilisent des nombres pour représenter des booléens. En Python, par exemple, **TRUE** correspond au nombre **1** et **FALSE** correspond au nombre **0**.

Si la <condition> est vérifiée, la valeur affectée est **1**, et la machine effectuera les instructions précisées ensuite.

Si la <condition> n'est pas vérifiée, la valeur affectée est 0, et la machine **n'effectuera pas** les instructions précisées ensuite.

Langage naturel

```
Si <condition>
  Alors <instructions>
FinSi
```

On peut **ajouter des instructions à exécuter dans le cas où la <condition> n'est pas vérifiée**, c'est-à-dire prévoir une structure du type ci-contre.

Langage naturel

```
Si <condition>
  Alors <instructions 1>
Sinon
  <instructions 2>
FinSi
```

Exemple : Un candidat est reçu à un examen si la moyenne M de ses notes est supérieure ou égale à 10. Pour afficher les résultats, on peut prévoir l'instruction :

Langage naturel

```
Si  $M \geq 10$ 
  Alors afficher « Candidat admis »
Sinon
  Afficher « Candidat recalé »
FinSi
```

Exercice 1 Tracer un symétrique Construire une figure

On veut écrire un algorithme permettant de tracer le symétrique $M'(x_{M'}; y_{M'})$ d'un point $M(x_M; y_M)$ par rapport à un point $O(x_O; y_O)$.

1 Écrire les coordonnées de M' en fonction de celles de M et de O .

$$O(x_O; y_O); O\left(\frac{x_M + x_{M'}}{2}; \frac{y_M + y_{M'}}{2}\right)$$

$$\begin{cases} x_O = \frac{x_M + x_{M'}}{2} \\ y_O = \frac{y_M + y_{M'}}{2} \end{cases} \text{ donc } \begin{cases} 2x_O - x_M = x_{M'} \\ 2y_O - y_M = y_{M'} \end{cases}$$

.... RAPPEL

Dans un repère $(O; I; J)$ du plan, soient $A(x_A; y_A)$ et $B(x_B; y_B)$ deux points distincts. Le point M , milieu du segment $[AB]$, a pour coordonnées : $\left(\frac{x_A + x_B}{2}; \frac{y_A + y_B}{2}\right)$.

2 a) Compléter l'algorithme suivant permettant de calculer les coordonnées de M' .

Langage naturel

Variables	$x_{M'}, y_{M'}, x_M, y_M, x_O, y_O$ nombres flottants
Entrées	Saisir x_M, y_M, x_O, y_O
Traitement	Affecter à $x_{M'}$ la valeur $2x_O - x_M$ Affecter à $y_{M'}$ la valeur $2y_O - y_M$
Sortie	Afficher $x_{M'}, y_{M'}$

b) Compléter le programme suivant dans le langage de programmation de votre calculatrice.

T.I. Basic

```
PROGRAM:SYMETRY
:Input "abscisse de O:",X
:Input "ordonnee de O:",Y
:Input "abscisse de M:",U
:Input "ordonnee de M:",V
:2*X-U→A
:2*Y-U→B
:Disp "les coordonnées de
M' sont:"
:Disp A,B
```

Basic Casio

```
=====SYMETRY=====
"Abscisse de O? "?→X
"Ordonnee de O? "?→Y
"Abscisse de M? "?→U
"Ordonnee de M? "?→V
"les coordonnées de M
, sont: "
2*X-U→A
2*Y-U→B
```



BESOIN D'AIDE?

On pourra nommer A la variable correspondant à l'abscisse de M' , et B la variable correspondant à l'ordonnée de M' .

c) Tester le programme pour $M(1; 2)$ et $O(4; 4)$. $M'(2,5; 3)$

3 Modifier l'algorithme précédent pour tracer le symétrique du point M par rapport à O dans la fenêtre « graphe » de la calculatrice.

T.I. Basic

```
Effécran
Efftable
[entrée des variables]

Pt-Aff(X,Y,3)
Pt-Aff(U,V,3)
Pt-Aff(A,B,3)
```

Basic Casio

```
Cls
[entrée des variables]

PlotOn X,Y
PlotOn U,V
PlotOn A,B
```



BESOIN D'AIDE?

• **EffTable** pour TI, **Cls** pour Casio efface l'écran du graphique.
• **Pt-Aff(abscisse,ordonnée)** pour TI, **PlotOn (abscisse, ordonnée)** pour Casio affiche un point dans la fenêtre graphique.

Exercice 2 Triangle isocèle ? Instruction conditionnelle

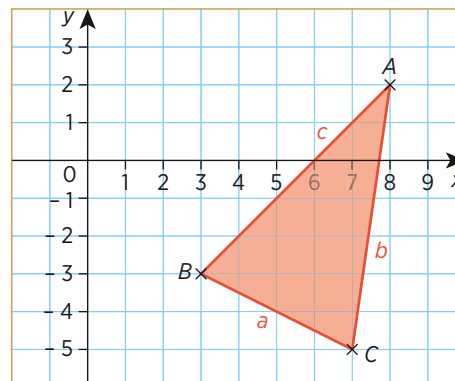
1 On veut écrire un programme permettant de déterminer si un triangle ABC est isocèle en A ou non, connaissant les coordonnées de ses trois sommets. On se place dans un repère orthonormé $(O ; I ; J)$ avec $O(0 ; 0)$, $I(1 ; 0)$ et $J(0 ; 1)$.

a) Les coordonnées de A , B et C sur la figure ci-contre sont :

$A(8 ; 2)$; $B(3 ; -3)$; $C(7 ; -5)$

b) Émettre une conjecture : en quel point le triangle semble-t-il être isocèle ?

Il semble être isocèle en A .



2 En utilisant les blocs ci-dessous, écrire un programme qui demande à l'utilisateur d'entrer les coordonnées des sommets d'un triangle ABC et qui indique si le triangle est isocèle en A ou non. (Indiquer dans le script « Bloc affectations » sans le recopier en entier.)

... RAPPEL

Il y a deux types d'affectations dans cet algorithme :

- On affecte une valeur à une variable lorsqu'on demande « abscisse de A » à l'utilisateur par exemple.

La valeur saisie par l'utilisateur est « enregistrée » dans la variable x_A .



- On affecte un calcul à une variable :

mettre a à $\sqrt{(x_C - x_B)^2 + (y_C - y_B)^2}$

Scratch

Quand « Drapeau vert » est cliqué

Bloc affectations

Mettre a à racine de $((x_C - x_B) * (x_C - x_B) + (y_C - y_B) * (y_C - y_B))$

Mettre b à racine de $((x_C - x_B) * (x_C - x_B) + (y_C - y_B) * (y_C - y_B))$

Mettre c à racine de $((x_C - x_B) * (x_C - x_B) + (y_C - y_B) * (y_C - y_B))$

Si $c = b$ alors

Dire Le triangle est isocèle en A pendant 2 secondes

Sinon

Dire Le triangle n'est pas isocèle en A pendant 2 secondes

3 Tester l'algorithme et vérifier la conjecture faite à la question 1.b).

Exercice 3

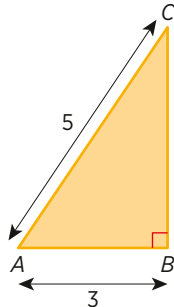
Calculer la longueur manquante dans un triangle rectangle

Instruction conditionnelle

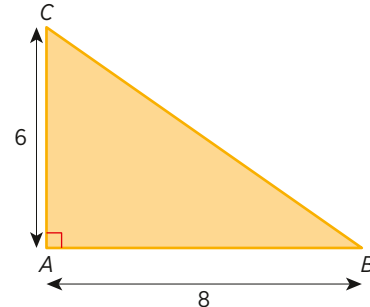
Pour la suite, on nomme les différentes longueurs d'un triangle ABC donné : $AC = X$; $AB = Y$; $BC = Z$.

1 Écrire un algorithme permettant de calculer la longueur manquante dans chacun des triangles rectangles suivants.

a)



b)



Langage naturel

Variables	X, Y, Z nombres flottants
Initialisation	Affecter à X la valeur 5 Affecter à Y la valeur 3
Traitement	Affecter à Z la valeur $\sqrt{5^2 - 3^2}$
Sortie	Afficher Z

Langage naturel

Variables	X, Y, Z nombres flottants
Initialisation	Affecter à X la valeur 6 Affecter à Y la valeur 8
Traitement	Affecter à Z la valeur $\sqrt{6^2 + 8^2}$
Sortie	Afficher Z

2 Écrire un algorithme dans le langage de programmation Python permettant à l'utilisateur de saisir les différentes longueurs des côtés d'un triangle ABC , et permettant d'afficher si ce triangle est rectangle ou non.

```
X=float(input("X="))
Y=float(input("Y="))
Z=float(input("Z="))
if X2=Y2+Z2 or Y2=X2+Z2 or Z2=X2+Y2
    print(« Le triangle est rectangle »)
else
    print(« Le triangle n'est pas rectangle »)
```

3 Modifier l'instruction conditionnelle dans l'algorithme précédent pour, lorsque le triangle est rectangle, afficher en quel point.

```
if X2=Y2+Z2
    print(« Le triangle est rectangle en B »)
elif Y2=X2+Z2
    print(« Le triangle est rectangle en C »)
elif Z2=X2+Y2
    print(« Le triangle est rectangle en A »)
else
    print(« Le triangle n'est pas rectangle »)
```

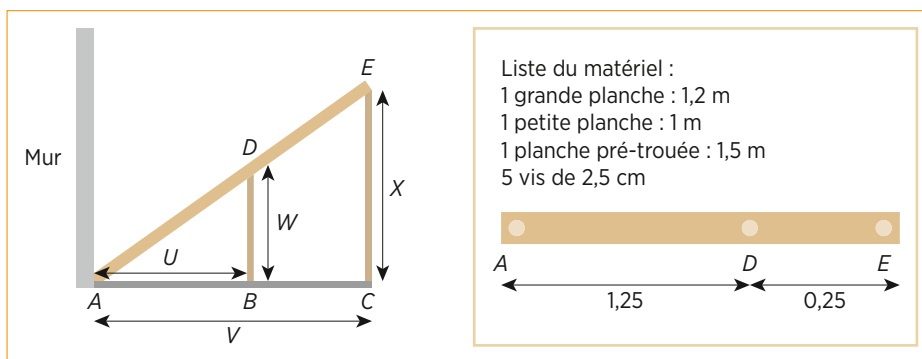


BESOIN D'AIDE ?

La structure de cet algorithme est :
Si ... Alors ...
Sinon si ... Alors ...
Sinon si ... Alors ...
Sinon
« Sinon si » s'écrit « elif » en Python (contraction de « else if »).

Exercice 4 Construire une étagère

Instruction conditionnelle



Julia est en train de monter son étagère Iquéa. Elle a oublié son niveau à bulle. Elle sait que la petite planche verticale de 1 m est perpendiculaire au sol et elle a fixé la planche inclinée pré-trouée en D et A. Elle aimerait vérifier que la grande planche est également perpendiculaire au sol avant de la visser à la planche inclinée. Elle connaît la distance entre le mur et la petite planche : $AB = 0,75$ m. On nomme les différentes longueurs des segments par une lettre : AB se nomme U , AC se nomme V , BD se nomme W , CE se nomme X .

- 1 Compléter l'algorithme ci-contre permettant de demander à l'utilisateur la longueur AC puis d'afficher si les droites (BD) et (CE) sont parallèles.

Langage naturel

Variables	U, V, W, X nombres flottants
Initialisation	Affecter à U la valeur 75 Affecter à W la valeur 100 Affecter à X la valeur 120
Entrées	Saisir V
Traitement	Si $\frac{W}{X} = \frac{U}{V}$ Alors afficher « (BD) et (CE) sont parallèles. » Sinon afficher « (BD) et (CE) ne sont pas parallèles. »
Fin de l'algorithme	

- 2 Expliquer comment modifier l'algorithme de la question précédente pour que l'utilisateur puisse entrer les différentes longueurs AB , AC , BD et CE .

On supprime les lignes Initialisation et on modifie la ligne Entrée : **Entrées** : Saisir U, W, X, V .

- 3 Julia trouve que cette méthode par tâtonnement n'est pas assez rapide et elle décide d'écrire un programme qui lui donnera directement la distance AC pour que les planches soient parallèles.

- a) A, B, C et A, D, E sont alignés dans cet ordre et distincts. En utilisant la réciproque du théorème de Thalès, écrire AC en fonction de AB , BD et CE :

$$AC = \frac{AB \times CE}{BD} \quad \text{Soit, en fonction de } U, W \text{ et } X, V = \frac{U \times X}{W}$$

- b) Compléter l'algorithme ci-contre.

Langage naturel

Variables	U, V, W, X nombres flottants
Entrées	Saisir U, W, X
Traitement	Affecter à V la valeur $\frac{U \times X}{W}$
Sortie	Afficher « (BD) est parallèle à (CE) si $AC = V$ » Afficher V



Gestion des séismes

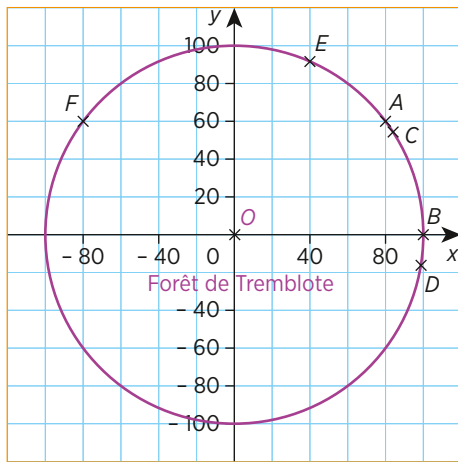
Instruction conditionnelle



La naissance d'un séisme a été détectée sur l'écran d'un sismographe (appareil qui enregistre les ondes permettant de localiser les séismes et de calculer leur magnitude). Les sismologues situent l'épicentre dans une petite forêt nommée Forêt de Tremblote. Ils vont devoir détecter les maisons se situant à moins de 100 km de l'épicentre pour pouvoir vérifier l'état des habitations les plus touchées.

Pour repérer les maisons à risque, ils introduisent un repère orthonormé sur leur écran de contrôle. Ils placent l'origine du repère O au niveau de l'épicentre du séisme et ils prennent le kilomètre pour unité.

Voici ce qui apparaît à l'écran :



Voici les coordonnées des maisons A, B, C, D, E et F calculées par les chercheurs :

Équation du cercle : $x^2 + y^2 = 1000$
 A(80 ; 60,1)
 B(100 ; 0)
 C(84 ; 56)
 D(100 ; -16)
 E(40 ; 92)
 F(-80 ; 60)
 O(0 ; 0)

1 On souhaite écrire un programme permettant de tester si une maison se situe à moins de 100 km du foyer.

a) Compléter l'algorithme ci-contre.

Langage naturel

Variables X, Y nombres flottants

Entrées Saisir X, Y

Traitement Si $\sqrt{X^2 + Y^2} \leq 100$

Alors afficher « Habitations à vérifier »

Sinon afficher « Habitations non touchées »

Fin de l'algorithme

b) Traduire le programme dans le langage de votre calculatrice et l'exécuter pour localiser, parmi les habitants des maisons A, B, C, D, E et F, lesquelles doivent être vérifiées.

T.I. Basic

```
PROGRAM: SEISME
:Input "Abscisse du point?"
",X
:Input "Ordonnée du point?"
",Y
:If  $\sqrt{X^2+Y^2} \leq 100$ 
:Then
:Disp "Habitations à vérifier"
:Else
:Disp "Habitations non touchées"
```

Basic Casio

```
=====SEISME =====
"Abscisse du point"?→
X↵
"Ordonnée du point"?→
Y↵
If  $\sqrt{X^2+Y^2} \leq 100$ 
Then "Habitations à vérifier"
Else "Habitations non touchées"
```

Il faut faire évacuer les habitants des maisons : B et F.

2 Écrire un algorithme permettant à l'utilisateur d'entrer les coordonnées du centre O d'un cercle, d'un point A par lequel passe le cercle, d'un point M quelconque, puis affiche si le point saisi appartient au cercle ou non. Le programme permet également de visualiser le cercle et les points O et M dans la fenêtre « graphe ».

a) Écrire l'algorithme en langage naturel.

- O est le centre du cercle, il a pour coordonnées $(O ; P)$.
- A est un point du cercle, il a pour coordonnées $(A ; B)$.
- M est un point quelconque, il a pour coordonnées $(M ; N)$.
- On nomme « R » le rayon du cercle.

Langage naturel

Variables	O, P, A, B, M, N nombres flottants
Entrées	Saisir O, P, A, B, M, N
Traitement	Affecter à R la valeur $\sqrt{(A-O)^2 + (B-P)^2}$
Sortie	Tracer Cercle (O, P, R)
	Afficher Point (O, P)
	Afficher Point (M, N)
	Si $(M-O)^2 + (N-P)^2 = R^2$
	Alors afficher « M sur le cercle »
	Sinon afficher « M non sur le cercle »

.... RAPPEL

Propriété : Soit r un réel strictement positif. Le point M appartient au cercle de centre O et de rayon r si et seulement si $OM = r$.



BESOIN D'AIDE ?

Écriture de la fonction cercle :
Pour T.I., Cercle(abscisse de O , ordonnée de O , rayon R) pour Casio, Circle(abscisse de O , ordonnée de O , rayon R).

b) Écrire l'algorithme de la question a) dans le langage de programmation de votre calculatrice.

T.I. Basic

```

EffTable
Prompt A,B
Prompt O,P
Prompt M,N

√((A-O)²+(B-P)²)→R
Cercle(O,P,R)
Pt-Aff(M,N,2,MAGENTA)
Pt-Aff(O,P,3,BLEU)
If (M-O)²+(N-P)²=R²
Then
Disp "M EST SUR LE CERCLE"
Else
Disp "M NON SUR LE CERCLE"
End

```

Basic Casio

```

Cls↓
?→A↓
?→B↓
?→O↓
?→P↓
?→M↓
?→N↓

√((A-O)²+(B-P)²)→R↓
Circle(O,P,R)↓
PlotOn(M,N)↓
PlotOn(O,P)↓
If (M-O)²+(N-P)²=R²↓
Then "M EST SUR LE CERCLE"
Else "M NON SUR LE CERCLE"
IfEnd

```

c) Exécuter le programme. Le tester pour les valeurs suivantes et écrire le résultat affiché.

$A = -2 ; B = -1 ; O = 0,5 ; P = 0,5 ; M = 2 ; N = 3$. Résultat : M est sur le cercle.

$A = -3 ; B = 2 ; O = 1 ; P = 0,3 ; M = 3 ; N = -2$. Résultat : M n'est pas sur le cercle.

d) Régler la « fenêtre », puis visualiser la figure dans « graphe ».

$X_{\min} = -6, X_{\max} = 6, \text{pas} = 1 ; Y_{\min} = -4, Y_{\max} = 4, \text{pas} = 1$.

Exercice 1 Appartenance d'un point à une droite

Instruction conditionnelle

On considère le script ci-contre écrit avec le logiciel Scratch, qui détermine si un point appartient ou non à une droite (d) donnée.

```

Scratch
quand est cliqué
demander Saisir l'abscisse du point et attendre
mettre X à réponse
demander Saisir l'ordonnée du point et attendre
mettre Y à réponse
si (-2 * X + 3 = Y) alors
  dire Le point appartient à la droite (d) pendant 2 secondes
sinon
  dire Le point n'appartient pas à la droite (d) pendant 2 secondes
  
```

1 Combien de variables (au sens algorithmique) ce script comporte-t-il ? Quels sont leurs noms ?

Ce script comporte deux variables, X et Y.

2 Quelle est l'équation de la droite (d) ?

$y = -2x + 3$.

3 Qu'affiche le script lorsqu'on saisit 2 pour valeur de X et -7 pour valeur de Y ?

Comme $-2 \times 2 + 3 = -1 \neq -7$, le script affichera « Le point n'appartient pas à la droite (d) ».

4 Qu'affiche le script lorsqu'on saisit -1,5 pour valeur de X et 6 pour valeur de Y ?

Comme $-2 \times (-1,5) + 3 = 6$, le script affichera « Le point appartient à la droite (d) ».

5 Lorsqu'on saisit 2 comme valeur de Y, quelle valeur de X faut-il saisir pour que le script affiche « Le point appartient à la droite (d) » ?

Il faut que $-2 \times X + 3 = 2$, soit $X = 0,5$.

Exercice 2 Droite passant par un point

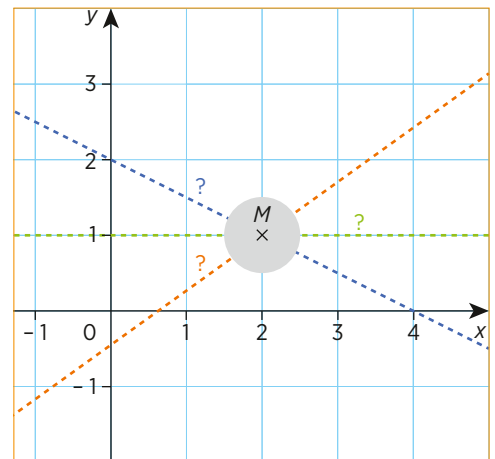
Instruction conditionnelle

On souhaite écrire un algorithme en langage naturel qui vérifie si une droite passe par un point donné, en l'occurrence le point M de coordonnées (2 ; 1).

1 Compléter l'algorithme ci-dessous pour qu'il affiche si la droite de coefficient directeur A et d'ordonnée à l'origine B passe par le point M ou non.

Langage naturel

Variables A et B sont des nombres flottants.....
Entrées Saisir A et B
Traitement Si $A \times 2 + B = 1$ alors
 Afficher « La droite passe par le point M. »
 Sinon Afficher « La droite ne passe pas par le point M. »
 FinSi



2 Qu'affiche l'algorithme lorsqu'on saisit 2 pour valeur de A et 1 pour valeur de B ?

Comme $2 \times 2 + 1 = 5 \neq 1$, l'algorithme affichera « La droite ne passe pas par le point M. »

- 3 Si on saisit 0 pour valeur de A , quelle valeur de B doit-on saisir pour que l'algorithme affiche « La droite passe par le point M . » ?

Il faut alors que $0 \times 2 + B = 1$, donc il faut saisir $B = 1$.

- 4 Il existe une droite passant par M mais pour laquelle on ne peut appliquer cet algorithme. Quelle est l'équation de cette droite ?

Cet algorithme ne fonctionne que pour des droites non verticales, dont l'équation est de la forme $y = ax + b$. Il s'agit donc de la droite verticale d'équation $x = 2$.

Exercice 3 Recherche de coefficient directeur

Variables et instructions élémentaires

On donne ci-dessous un algorithme, traduit dans trois langages de programmation, permettant de calculer le coefficient directeur d'une droite d'équation $y = ax + b$, passant par les points de coordonnées $(X_1; Y_1)$ et $(X_2; Y_2)$.

Python

```
X=float(input("X1="))
Y=float(input("Y1="))
Z=float(input("X2="))
T=float(input("Y2="))
A=(T-Y)/(Z-X)
print("A=",A)
```

T.I. Basic

```
PROGRAM:COEFFDIR
:Input "X1",X
:Input "Y1",Y
:Input "X2",Z
:Input "Y2",T
:(T-Y)/(Z-X)→A
:Disp A
```

Basic Casio

```
====COEFFDIR=====
"X1=": ?→X#
"Y1=": ?→Y#
"X2=": ?→Z#
"Y2=": ?→T#
(T-Y)÷(Z-X)→A#
A#
```

- 1 Combien de variables (au sens algorithmique) ce script comporte-t-il ? Quels sont leurs noms ? Quelle est celle correspondant au coefficient directeur ?

Cinq variables apparaissent dans ce script : X, Y, Z, T et A . C'est la variable A qui contient la valeur du coefficient directeur, grâce au calcul $\frac{Y_2 - Y_1}{X_2 - X_1}$ en ligne 5.

- 2 Implanter cet algorithme dans le langage de votre choix. Qu'affiche-t-il avec les points de coordonnées $(-1; 2)$ et $(2; 8)$?

Il affiche un coefficient directeur égal à 2.

- 3 Quel problème pose cet algorithme si on saisit la même valeur pour X_1 et X_2 ? À quel type de droite cela correspond-il ?

Une division par 0 apparaît dans le calcul. La droite est alors verticale, d'équation du type $x = X_1$.

- 4 Compléter l'algorithme ci-dessous pour qu'il affiche le type de droite si les deux abscisses saisies sont égales, et affiche le coefficient directeur si ce n'est pas le cas.

Langage naturel

Variables	X, Y, Z, T et A sont des nombres flottants
Entrées	Saisir X, Y, Z et T
Traitement	Si $X = Z$ alors
	Afficher « La droite est verticale. »
	Sinon Affecter à A la valeur $(T - Y)/(Z - X)$
	Afficher « Le coefficient directeur est égal à : »
	Afficher A
	FinSi

Exercice 4

Calcul des coordonnées du point d'intersection de deux droites

Notion de fonction

On souhaite déterminer les coordonnées du point d'intersection M des deux droites d'équations $y = 2x + 1,5$ et $y = -2x + 2,5$. Ses coordonnées devant vérifier les deux équations, on en déduit que son abscisse est la solution de l'équation $2x + 1,5 = -2x + 2,5$.

- 1 Résoudre cette équation et déterminer les coordonnées du point d'intersection.

$2x + 1,5 = -2x + 2,5 \Leftrightarrow 4x = 1 \Leftrightarrow x = 0,25$, puis $y = 2 \times 0,25 + 1,5 = 2$. Donc $M(0,25 ; 2)$.

- 2 Dans le cas général, avec deux droites d'équations $y = ax + b$ et $y = a'x + b'$, exprimer x en fonction de a , b , a' et b' .

On obtient, en résolvant l'équation $ax + b = a'x + b'$, que $x = \frac{b' - b}{a - a'}$.

- 3 Compléter maintenant, en langage Python, la fonction (au sens algorithmique) qui prend en arguments les coefficients directeurs et ordonnées à l'origine des deux droites et qui retourne les coordonnées du point d'intersection sous la forme d'une liste.

Python

```
def intersection(a,a',b,b') :
    X= (b'-b)/(a-a')
    Y= a*X+b
    M=[X,Y]
    return M
```



BESOIN D'AIDE?

Une variable de type **liste** permet de contenir plusieurs valeurs. En Python, on écrit ces valeurs **entre crochets** pour définir une liste. Ensuite, on peut récupérer chaque valeur par son indice (sa position), en partant de 0. Par exemple, si une liste **L** contient 3 valeurs, la première est **L[0]**, la deuxième **L[1]** et la troisième **L[2]**.

- 4 Modifier la fonction précédente pour qu'elle prenne en arguments deux listes D1 et D2. Chaque liste possédera deux valeurs : la première étant le coefficient directeur d'une droite et la deuxième son ordonnée à l'origine.

Python

```
def intersection(D1,D2) :
    X= (D2[1]-D1[1])/(D1[0]-D2[0])
    Y= D1[0]*X+D1[1]
    M= [X,Y]
    return M
```

Exercice 5

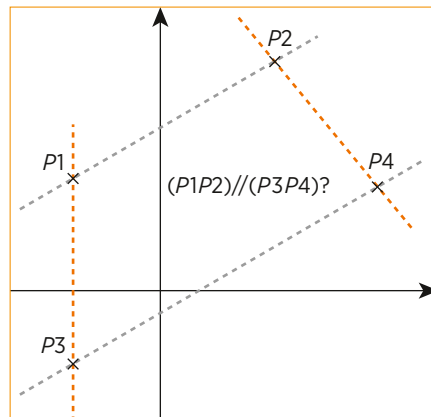
Parallélisme et alignement de points

Notion de fonction et listes

Étant donnés quatre points du plan, on souhaite établir si la droite passant par les deux premiers est parallèle à celle passant par les deux derniers.

- 1 Pour stocker les coordonnées de ces points, de combien de variables de type flottant a-t-on besoin ? Et si on utilise une liste par point ?

On aura besoin de $4 \times 2 = 8$ variables de type flottant, ou bien seulement de 4 listes contenant deux nombres flottants chacune.



2 On va donc définir une fonction (au sens algorithmique) qui prend comme arguments les coordonnées de deux points P1 et P2, et qui retourne le coefficient directeur de la droite passant par ces deux points ou bien le texte « V » si la droite est verticale. Compléter la fonction ci-contre en pseudo-langage.



BESOIN D'AIDE?

Cette fonction retourne une valeur de **type** différent selon les valeurs données en arguments : soit une **chaîne de caractères**, soit un **nombre flottant**.

3 Traduire cette fonction en langage Python en complétant le script ci-contre.

4 On a maintenant besoin de saisir les coordonnées des quatre points. Plutôt que de répéter plusieurs fois les mêmes instructions, on crée une fonction permettant cette saisie et qui retourne une liste contenant les coordonnées d'un point.

Compléter le script Python suivant, pour qu'il saisisse quatre points (variables P1, P2, P3 et P4, sous forme de listes) et qu'il affiche si les droites (P1P2) et (P3P4) sont parallèles ou non, en utilisant la fonction `coeffdir`.

Python

```
# Calcul du coefficient directeur
def coeffdir(P1,P2):
    DX=P2[0]-P1[0]
    DY=P2[1]-P1[1]
    if DX==0:
        return "V"
    else:
        A=DY/DX
        return A

# Saisie des points
def saisie():
    X=float(input("Entrer l'abscisse: "))
    Y=float(input("Entrer l'ordonnée: "))
    return [X,Y]

# Main
P1=saisie()
P2=saisie()
P3=saisie()
P4=saisie()

if coeffdir(P1, P2)==coeffdir(P3,P4) :
    print("Les droites sont parallèles.")
else:
    print("Les droites ne sont pas parallèles.")
```

Pseudo-langage

```
Fonction coeffdir(P1,P2)
    DX←P2[0] - P1[0]
    DY←P2[1] - P1[1]
    Si DX = 0
        Retourner « V »
    Sinon
        A←DY/DX
        Retourner A
    Fin Si
Fin Fonction
```

Python

```
def coeffdir(P1,P2):
    DX= P2[0]-P1[0]
    DY= P2[1]-P1[1]
    if DX==0:
        return "V"
    else:
        A=DY/DX
        return A
```

Python

```
def saisie():
    X=float(input("Entrer l'abscisse: "))
    Y=float(input("Entrer l'ordonnée: "))
    return [X,Y]
```



BESOIN D'AIDE?

En algorithmique, une fonction peut ne pas prendre d'argument. On l'appelle simplement avec des **parenthèses vides**.

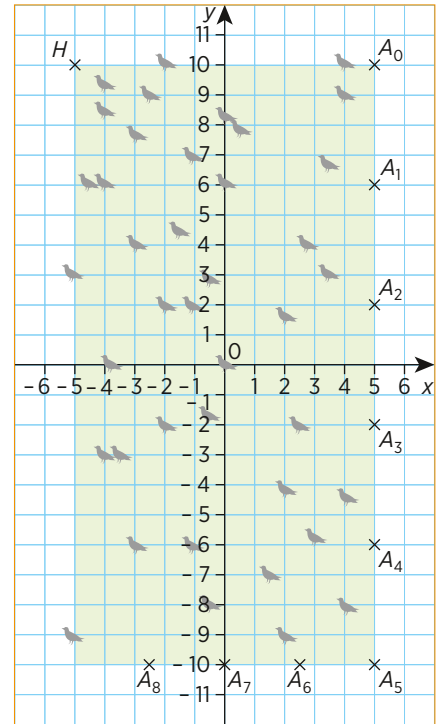
5 Comment utiliser le script précédent pour vérifier l'alignement de trois points ?

Deux droites parallèles qui ont un point commun sont confondues. Pour savoir si trois points sont alignés, il suffit donc d'appliquer ce script en saisissant un même point pour les deux droites, par exemple P1 et P3.



Balayage Boucle bornée et notion de fonction

La carte ci-contre représente le lac Stymphale, au bord duquel Hercule (le point H) se positionne afin de tuer les oiseaux cruels y résidant. Il ne dispose que d'une seule flèche, mais son tir est si puissant qu'il pourra atteindre tous les oiseaux se trouvant sur sa trajectoire. Pour viser, Hercule doit cependant obligatoirement atteindre l'une des 9 cibles représentées par les points A_i (pour i allant de 0 à 8) disposées le long du lac. On veut écrire un programme qui détermine la cible qui définit la trajectoire permettant à Hercule d'atteindre le maximum d'oiseaux.



Langage naturel

- Pour chaque cible A_i :
- Déterminer l'équation de la droite (HA_i)
 - Parcourir l'ensemble des points repérant les oiseaux et déterminer s'ils appartiennent à cette droite
 - Afficher le nombre d'oiseaux atteints par cette trajectoire

PARTIE 1 Détermination de l'équation d'une trajectoire

1 On considère la droite (HA_2). Calculer son coefficient directeur, puis à l'aide des coordonnées du point H , qui appartient à cette droite, déterminer son ordonnée à l'origine.

Les coordonnées de H sont $(-5 ; 10)$ et celles de A_2 sont $(5 ; 2)$. On obtient que le coefficient directeur de (HA_2) est égal à $-0,8$ et son équation est de la forme $y = -0,8x + b$. En utilisant les coordonnées de H , on obtient $10 = -0,8 \times (-5) + b$, soit $b = 6$.

2 D'une façon générale, toujours à l'aide des coordonnées de H , exprimer l'ordonnée à l'origine d'une droite (HA_i) en fonction de son coefficient directeur.

De son équation $y = ax + b$, on a $b = y - ax$, puis $b = 10 + 5a$ en remplaçant x et y par les coordonnées de H .

PARTIE 2 Détermination du nombre d'oiseaux situés sur une trajectoire

Dans cette partie, on commence par compter le nombre d'oiseaux atteints par la première trajectoire uniquement, c'est-à-dire le nombre de points situés sur la droite (HA_0). On a donc besoin d'une liste des coordonnées des points, nommée listeOiseaux, et d'une fonction coeffdir pour calculer le coefficient directeur des droites. Il faut également compter les points situés sur (HA_0), en utilisant une variable « compteur » que l'on nomme N .

Script 1 Python

```
def coeffdir(P1,P2):
    DX=P2[0]-P1[0]
    DY=P2[1]-P1[1]
    if DX==0:
        return "V"
    else:
        A=DY/DX
        return A

listeOiseaux=[[-5,-9],[-5,3],[-4.5,6],[-4,9.2],[-4,8.4],
               [-4,6],[-4,-3],[-3.75,0],[-3.5,-3],[-3,7.6],
               [-3,-6],[-3,4],[-2.5,9],[-2,10],[-2,2.1],
               [-2,-2],[-1.5,4.4],[-1,6.8],[-1,2],[-1,-6],
               [-0.5,2.8],[-0.5,-1.8],[-0.4,-8],[0,6],[0,0],
               [0,8.2],[0.5,7.8],[1.5,-7],[2,1.6],[2,-4.2],
               [2,-9],[2.5,-2],[2.75,4],[3,-5.8],[3.5,6.6],
               [3.5,3],[4,10],[4,-4.4],[4,-8],[4,9]]
```

3 Quel est le type de la variable N ? Quelle instruction écrire pour augmenter sa valeur de 1 (on dit qu'on incrémente N de 1) ?

N est un nombre entier. On écrira $N = N + 1$.

4 Compléter le début du programme suivant en :

- initialisant les variables H et $A0$ comme des listes contenant les coordonnées des points H et $A0$;
- initialisant la variable A en utilisant la fonction `coeffdir` et la variable N à 0 ;
- écrivant le test d'appartenance du point « oiseau » à la droite qui détermine s'il faut incrémenter N .

Python

```
H= [-5,10]
A0= [5,10]
A= coeffdir(H, A0)
B= 10+5*A
N= 0

for oiseau in listeOiseaux:
    if A*oiseau[0]+B == oiseau[1] :
        N=N+1
print(N)
```



BESOIN D'AIDE ?

- La boucle **for** permet de parcourir la liste **listeOiseaux** élément par élément. Autrement dit, la variable **oiseau** va prendre successivement toutes les valeurs contenues dans **listeOiseaux**, c'est-à-dire les coordonnées des points, sous forme elles-mêmes de listes.
- Pour récupérer l'abscisse du point **oiseau**, on utilisera **oiseau[0]** et pour l'ordonnée, **oiseau[1]**.

5 À l'aide de ce programme, déterminer le nombre N d'oiseaux situés sur la droite (HA_0) . Vérifier ce résultat sur la carte.

Le programme retourne $N = 2$. On voit sur la carte qu'il y a deux oiseaux sur la droite (HA_0) , ce qui est confirmé par deux points d'ordonnée 10 dans la liste $([-2 ; 10]$ et $[4 ; 10])$.

PARTIE 3 Balayage des cibles

Maintenant qu'on sait récupérer le nombre d'oiseaux sur chaque trajectoire, il suffit de le faire pour chacune d'elles, c'est-à-dire chaque droite (HA_i) .

6 Quelle est la seule variable du script précédent qui doit changer lors de sa répétition pour chaque droite ?

Seul le point cible (A_i) doit changer.

7 Pour pouvoir écrire une boucle **for** qui parcourt les cibles A_i , il faut créer une liste contenant leurs coordonnées. Compléter la liste « listeCibles » suivante, sachant que les coordonnées des points A_0 et A_1 y ont été indiquées.

listeCibles=[[5,10],[5,6],[5,2],[5,-2],[5,-6],[5,-10],[2,5,-10],[0,-10],[2,5,-10]]

8 Quelle est la longueur de cette liste, ou autrement dit, combien de fois va-t-on répéter le script de la partie 2 ?

Cette liste contient 9 valeurs. On va donc répéter le script 9 fois.

9 Compléter le script 1, en sachant que maintenant le point A_i est stocké dans la liste `listeCibles`, en position i , c'est-à-dire `listeCibles[i]`.

Suite du script 1

```
listeCibles=[[5,10],[5,6],[5,2],[5,-2],[5,-6],[5,-10]
             [2,5,-10],[0,-10],[2,5,-10],[5,-10]]

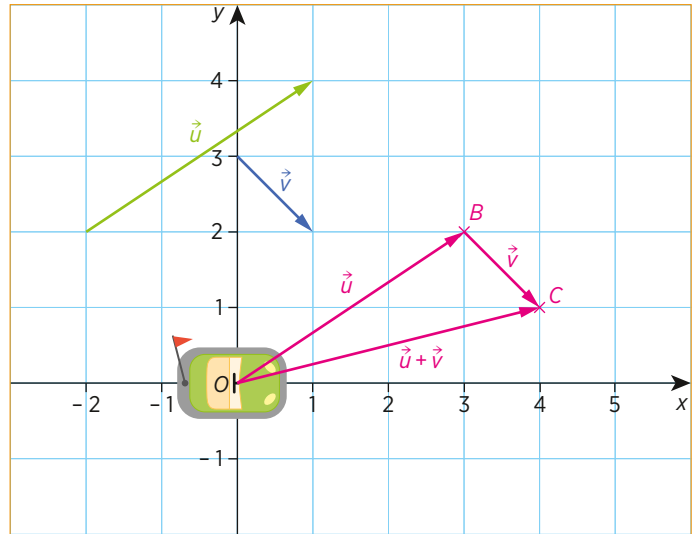
H= [-5,10]

for i in range(9):
    A= coeffdir(H,listeCibles[i])
    B= 10+5*A
    N= 0
    print("Cible A",i)
    for oiseau in listeOiseaux:
        if A*oiseau[0]+B == oiseau[1] :
            N= N+1
    print(N)
```

Exercice 1 Les auto-tamponneuses

Variables et instructions élémentaires

Ophélie est au volant de son auto-tamponneuse. Elle se trouve au milieu de la piste, en O , origine d'un repère orthonormé, quand une autre auto la percute sur le côté. Elle effectue une première translation de vecteur \vec{u} puis rebondit sur l'auto de Bilal en B pour repartir selon une translation de vecteur \vec{v} et s'arrêter en C .



1 Positionner les points B et C sur le schéma.

2 Quel vecteur permet une translation directe du point O au point C ? En tracer un représentant.

Le vecteur $\vec{u} + \vec{v}$.

3 Déterminer par lecture graphique les coordonnées des vecteurs \vec{u} , \vec{v} et $\vec{u} + \vec{v}$.

$\vec{u} \begin{pmatrix} 3 \\ 2 \end{pmatrix}$, $\vec{v} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ et $(\vec{u} + \vec{v}) \begin{pmatrix} 4 \\ 1 \end{pmatrix}$.

4 Compléter l'algorithme ci-contre permettant d'obtenir les coordonnées du point d'arrivée $C(x_C ; y_C)$ de l'auto d'Ophélie lorsqu'elle effectue deux translations successives de vecteurs $\vec{u} \begin{pmatrix} x \\ y \end{pmatrix}$ et $\vec{v} \begin{pmatrix} x' \\ y' \end{pmatrix}$ donnés par l'utilisateur, en partant de l'origine du repère O .

Langage naturel

Variables	x, y, x', y', x_C, y_C nombres flottants
Entrée	Saisir x, y, x', y'
Traitement	Affecter à x_C la valeur $x + x'$ Affecter à y_C la valeur $y + y'$
Sortie	Afficher « Les coordonnées du point C sont : » Afficher les variables x_C, y_C

5 Écrire l'algorithme dans le langage de programmation de votre calculatrice et le tester. Indiquer quelles sont les coordonnées du point C lorsque $\vec{u} \begin{pmatrix} 1 \\ 5 \end{pmatrix}$ et $\vec{v} \begin{pmatrix} -3 \\ -6 \end{pmatrix}$: $C(-2 ; -1)$.

T.I. Basic

Prompt A, B, C, D
 $A+C \rightarrow E$
 $B+D \rightarrow F$
 Disp "LES COORDONNEES DE C SONT :"
 Disp E,F
 End

Basic Casio

"A=" \rightarrow A
 "B=" \rightarrow B
 "C=" \rightarrow C
 "D=" \rightarrow D
 "Les coordonnees de C sont :"
 $A+C \rightarrow E$
 $B+D \rightarrow F$

Exercice 2 Lieu mystère Boucle bornée

Zahra est en vacances d'hiver, elle a envie d'envoyer un message original à ses amis. Voici son message envoyé par mail :

Langage naturel

Variables a, b, x, y, x', y' nombres flottants
Entrée Saisir x, y, x', y'
Initialisation a prend la valeur 0
 b prend la valeur 0
Traitement Pour i allant de 1 à 3
 Affecter à a la valeur $a + x$
 Affecter à b la valeur $b + y$
 Tracer le segment entre les points de coordonnées $(a ; b)$ et $(a + x ; b + y)$
 Affecter à a la valeur $a + x'$
 Affecter à b la valeur $b + y'$
 Tracer le segment entre les points de coordonnées $(a ; b)$ et $(a + x' ; b + y')$
 Fin Pour
 Affecter à a la valeur $a + 6x'$
 Affecter à b la valeur $b + 6y'$
 Tracer le segment entre les points de coordonnées $(a ; b)$ et $(a + x' ; b + y')$

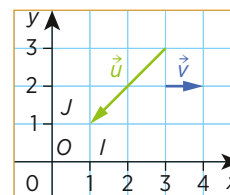
Fin de l'algorithme

« Hello ! Pour savoir si je suis à la mer ou à la montagne, programme l'algorithme suivant sur ta calculatrice ou avec Python.

PS : on considère deux vecteurs $\vec{u} \begin{pmatrix} x \\ y \end{pmatrix}$ et $\vec{v} \begin{pmatrix} x' \\ y' \end{pmatrix}$ et un point $M(a ; b)$ dans le repère orthonormé $(O ; I ; J)$. »

1 a) Calculer les coordonnées des vecteurs suivants.

Les coordonnées des vecteurs \vec{u} et \vec{v} sont : $\vec{u}(-2 ; -2)$ et $\vec{v}(1 ; 0)$.



b) Écrire l'algorithme de l'énoncé dans le langage de programmation de votre calculatrice. Effectuer les réglages indiqués dans la fenêtre « graphe » de la calculatrice.

$X_{\min} = -15$, $X_{\max} = 0$, pas = 1 ;
 $Y_{\min} = -7$, $Y_{\max} = 1$, pas = 1.



BESOIN D'AIDE ?

TI Pour tracer un segment entre deux points M et N on utilise la fonction Ligne :
 $2^{\text{nde}} \rightarrow \text{prgr} \rightarrow 2:\text{Ligne}(\text{abscisse de } M, \text{ordonnée de } M, \text{abscisse de } N, \text{ordonnée de } N)$

Casio Pour tracer un segment entre deux points M et N on utilise la fonction Ligne :
 $2^{\text{nde}} \rightarrow \text{Catalogue} \rightarrow \text{F-Line}$ abscisse de M , ordonnée de M , abscisse de N , ordonnée de N

T.I. Basic

Prompt A, B, C, D
 $0 \rightarrow X$
 $0 \rightarrow Y$
 EffDess
 For(I,1,3)
 Ligne(X,Y,A+X,B+Y,VERT)
 $A+X \rightarrow X$
 $B+Y \rightarrow Y$
 Ligne(X,Y,C+X,D+Y,VERT)
 $C+X \rightarrow X$
 $D+Y \rightarrow Y$
 End
 Ligne(X,Y,6C+X,6D+Y, VERT)

Basic Casio

"A=" \rightarrow A,↓
 "B=" \rightarrow B,↓
 "C=" \rightarrow C,↓
 "D=" \rightarrow D,↓
 $0 \rightarrow X,↓$
 $0 \rightarrow Y,↓$
 ClrGraph,↓
 For 1 \rightarrow I To 3,↓
 F-Line X,Y,A+X,B+Y,↓
 $A+X \rightarrow X,↓$
 $B+Y \rightarrow Y,↓$
 F-Line X,Y,C+X,D+Y,↓
 Next,↓
 F-Line X,Y,6C+X,6D+Y

2 Où se trouve Zahra ?

Le dessin tracé est un sapin. Zahra se trouve à la montagne.

Exercice 3

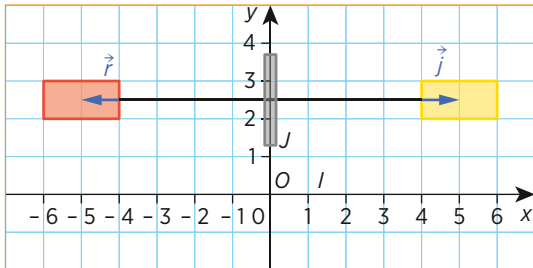
Tir à la corde

Instruction conditionnelle

Deux équipes s'affrontent au jeu de la corde. L'équipe des rouges est face à l'équipe des jaunes. Chaque équipe tire sur la corde, le but étant de faire passer l'équipe adverse de son côté de la ligne centrale.

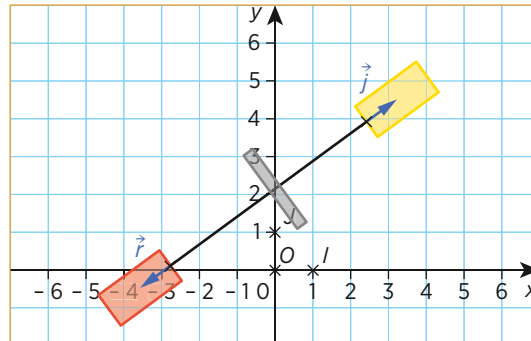
L'équipe rouge exerce sur la corde une force \vec{r} et l'équipe jaune exerce sur la corde une force \vec{j} .

Cas 1



norme de $\vec{r} <$ norme de \vec{j} : l'équipe jaune gagne.
norme de $\vec{r} >$ norme de \vec{j} : l'équipe rouge gagne.

Cas 2



1 a) Dans le Cas 1 décrit ci-dessus, que valent y et y' ? $y = 0$ et $y' = 0$.

b) Compléter l'algorithme ci-contre indiquant l'équipe gagnante en fonction des vecteurs

$$\vec{r} \begin{pmatrix} x \\ y \end{pmatrix} \text{ et } \vec{j} \begin{pmatrix} x' \\ y' \end{pmatrix}.$$

c) Exécuter l'algorithme et déterminer l'équipe gagnante lorsque $\vec{r} \begin{pmatrix} 3 \\ 0 \end{pmatrix}$ et $\vec{j} \begin{pmatrix} -5 \\ 0 \end{pmatrix}$.

C'est l'équipe jaune.

Langage naturel

Variables	$x, x',$ nombres flottants
Entrée	Saisir x, x'
Traitement	Si $\sqrt{x^2} < \sqrt{x'^2}$ Alors afficher « L'équipe jaune gagne. » Sinon Afficher « L'équipe rouge gagne. »
Fin de l'algorithme	

2 a) On se place dans une situation similaire à celle du Cas 2, les vecteurs \vec{r} et \vec{j} ont la même direction, sont de sens contraire et ont des normes différentes. Compléter l'algorithme ci-contre indiquant l'équipe gagnante en fonction des vecteurs

$$\vec{r} \begin{pmatrix} x \\ y \end{pmatrix} \text{ et } \vec{j} \begin{pmatrix} x' \\ y' \end{pmatrix}.$$

Langage naturel

Variables	x, y, x', y' nombres flottants
Entrée	Saisir x, y, x', y'
Traitement	Si $\sqrt{x^2 + y^2} < \sqrt{x'^2 + y'^2}$ Alors afficher « L'équipe jaune gagne. » Sinon Afficher « L'équipe rouge gagne. »
Fin de l'algorithme	



BESOIN D'AIDE?

Les vecteurs ayant même direction et des sens opposés, c'est la différence entre les normes de chacun de ces vecteurs qui va nous permettre de déterminer l'équipe gagnante.

●●● RAPPEL

Dans un repère orthonormé, la distance entre deux points $A(x_A; y_A)$ et $B(x_B; y_B)$ est donnée par $AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$.

b) Exécuter l'algorithme sans machine et déterminer l'équipe gagnante lorsque $\vec{r} \begin{pmatrix} -6 \\ 3 \end{pmatrix}$ et $\vec{j} \begin{pmatrix} 3 \\ -1,5 \end{pmatrix}$.

$$\text{norme de } \vec{r} = \sqrt{(-6)^2 + 3^2} = 3\sqrt{5}$$

$$\text{norme de } \vec{j} = \sqrt{3^2 + (-1,5)^2} = \frac{3\sqrt{5}}{2} < 3\sqrt{5}$$

C'est l'équipe rouge qui gagne.

Exercice 4 Trajet en ville

Instruction conditionnelle et boucle bornée

Magalie fait ses courses en centre ville. Elle part de sa maison et se rend successivement dans 3 boutiques en effectuant des translations de vecteurs \vec{u} , \vec{v} , \vec{w} , \vec{x} .

- 1 Compléter l'algorithme ci-contre permettant de rentrer les coordonnées des 4 vecteurs et de savoir si Magalie revient à la maison après avoir fait ses courses.

Langage naturel

Variables	$x, y, x', y', x'', y'', x''', y''', x''', y'''$ nombres flottants
Entrée	Saisir $x, y, x', y', x'', y'', x''', y''', x''', y'''$
Traitement	Si $x + x' + x'' + x''' = 0$ et $y + y' + y'' + y''' = 0$ Alors afficher « Magalie rentre à la maison. » Sinon Afficher « Magalie ne rentre pas à la maison. »
Fin de l'algorithme	

- 2 Écrire l'algorithme dans le langage de programmation de votre calculatrice.

T.I. Basic

```
Prompt A, B, C, D, E, F, G, H
If A+B+C+D=0 et E+F+G+H=0
Then
Disp "Magalie rentre à la maison"
Else
Disp "Magalie ne rentre pas"

```

Basic Casio

```
"A="→A↓
[...↓
"H="→H↓
If A+B+C+D=0 et E+F+G+H=0↓
Then↓
"Magalie rentre à la maison"
Else
"Magalie ne rentre pas"
IfEnd

```

- 3 a) Exécuter le programme et dire, lorsque Magalie effectue les déplacements suivants :

$\vec{u} \begin{pmatrix} -6 \\ 3 \end{pmatrix}$, $\vec{v} \begin{pmatrix} 4 \\ 2 \end{pmatrix}$, $\vec{w} \begin{pmatrix} 1 \\ -5 \end{pmatrix}$, $\vec{x} \begin{pmatrix} 5 \\ 0 \end{pmatrix}$, si elle rentre à la maison après ses courses : non.

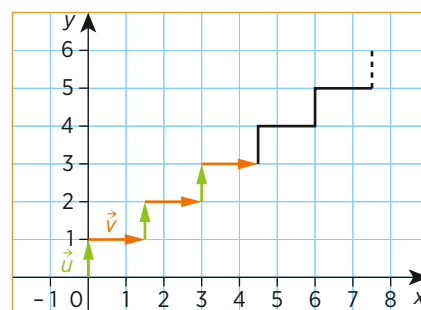
- b) Même question pour : $\vec{u} \begin{pmatrix} 8 \\ 2 \end{pmatrix}$, $\vec{v} \begin{pmatrix} 1 \\ -5 \end{pmatrix}$, $\vec{w} \begin{pmatrix} -9 \\ 7 \end{pmatrix}$, $\vec{x} \begin{pmatrix} 0 \\ -4 \end{pmatrix}$: oui.

Exercice 5 La fourmi dans l'escalier

Variables et instructions élémentaires

Une fourmi gravit les marches d'un escalier. On place l'escalier dans un repère orthonormé $(O; I; J)$.

La fourmi effectue un mouvement de translation de vecteur $\vec{u} \begin{pmatrix} 0 \\ y \end{pmatrix}$ suivi d'une translation de vecteur $\vec{v} \begin{pmatrix} x' \\ 0 \end{pmatrix}$.



1 Trouver par le calcul le nombre de marches que la fourmi doit gravir pour arriver au point de coordonnées (7,5 ; 5) lorsque $y = 1$ et $x' = 1,5$.

Gravir une marche correspond à un déplacement de vecteur $(\vec{u} + \vec{v})$ de coordonnées $\begin{pmatrix} 1,5 \\ 1 \end{pmatrix}$.

On cherche le nombre de marches k tel que $k(\vec{u} + \vec{v}) = \begin{pmatrix} 7,5 \\ 5 \end{pmatrix} \Leftrightarrow k \begin{pmatrix} 1,5 \\ 1 \end{pmatrix} = \begin{pmatrix} 7,5 \\ 5 \end{pmatrix}$

$$\begin{cases} k \times 1,5 = 7,5 \\ k \times 1 = 5 \end{cases} \Leftrightarrow \begin{cases} k = \frac{7,5}{1,5} = 5 \\ k = \frac{5}{1} = 5 \end{cases}$$

La fourmi doit gravir 5 marches.

2 Calculer les coordonnées du point P auquel arrive la fourmi après avoir gravi 8 marches lorsque $y = 2$ et $x' = 3$.

Le vecteur $(\vec{u} + \vec{v})$ a pour coordonnées $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$. On cherche les coordonnées du vecteur \vec{w} tel que $\vec{w} = 8(\vec{u} + \vec{v})$. Les coordonnées de \vec{w} sont donc $\vec{w} \begin{pmatrix} 24 \\ 16 \end{pmatrix}$ ($8 \times 3 = 24$ et $8 \times 2 = 16$).

$\vec{OP} = \vec{w}$, donc P a pour coordonnées (24 ; 16).

3 On veut écrire un algorithme indiquant les coordonnées du point d'arrivée de la fourmi $M(x_M ; y_M)$ lorsque celle-ci gravit k marches, avec k un nombre entier.

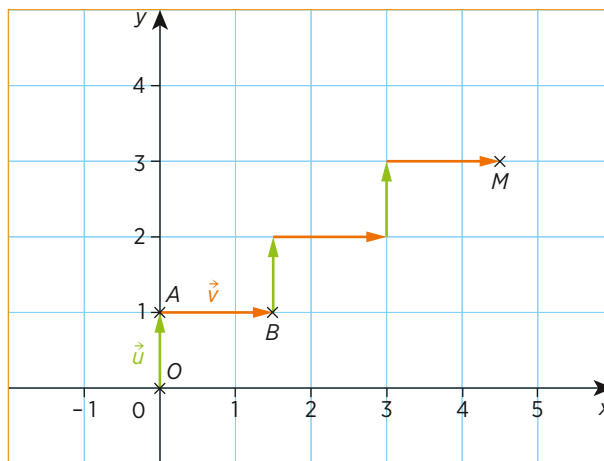
a) Écrire le vecteur \vec{OB} en fonction des vecteurs \vec{OA} et \vec{AB} (relation de Chasles), puis en fonction de \vec{u} et \vec{v} .

$$\vec{OB} = \vec{OA} + \vec{AB} = \vec{u} + \vec{v} ; \vec{OB} \begin{pmatrix} x' \\ y \end{pmatrix}$$

b) Écrire le vecteur \vec{OM} en fonction du nombre k et du vecteur \vec{OB} , puis en fonction du nombre k et des vecteurs \vec{u} et \vec{v} .

$$\vec{OM} = k \times \vec{OB} = k \times (\vec{u} + \vec{v}) ; \vec{OM} \begin{pmatrix} k \times x' \\ k \times y \end{pmatrix}$$

c) Compléter l'algorithme ci-contre.



Langage naturel

Variables	y, x', k, x_M, y_M nombres flottants
Entrée	Saisir y, x', k
Traitement	Affecter à x_M la valeur $k \times x'$ Affecter à y_M la valeur $k \times y$
Sortie	Afficher « Les coordonnées du point M sont : », x_M, y_M

4 Écrire l'algorithme dans le langage de programmation Python et l'exécuter. Compléter le tableau suivant.

y	x'	k	$M(x_M ; y_M)$
1	1	8	$M(8 ; 8)$
2	3,5	50	$M(175 ; 100)$
2	2,5	36	$M(90 ; 72)$
3,1	3,6	11	$M(39,6 ; 34,1)$

Python

```
y=float(input("y="))
x'=float(input("x'="))
k=float(input("k="))
xM=k*x'
yM=k*y
print("Les coordonnées de M sont :",xM,yM)
```



Navigation en mer

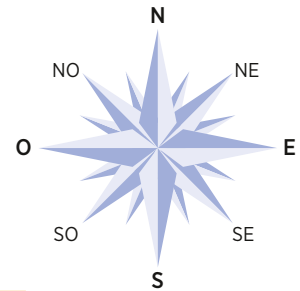
Instruction conditionnelle et boucle bornée

Deux bateaux naviguent sur la mer Méditerranée. Le centre de sécurité maritime connaît les déplacements prévus par les deux capitaines et veut s'assurer que les deux bateaux ne peuvent entrer en collision. Dans la zone de navigation on a matérialisé un repère orthonormé $(O ; I ; J)$ à l'aide de bouées.

Voici les feuilles de route de chacun des bateaux :

La Sardine Rose
Déplacement de direction :
Est - Est - Nord

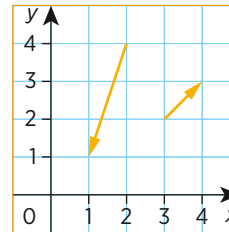
Le Calamar Doré
Déplacement de direction :
Est - Est
Puis Est - Est - Nord - Nord



Ce qui se traduit par :

Le vecteur de coordonnées $\begin{pmatrix} -1 \\ -3 \end{pmatrix}$ correspond au mouvement Ouest - Sud - Sud - Sud pour les marins.

Le vecteur de coordonnées $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ correspond au mouvement Est - Nord pour les marins.

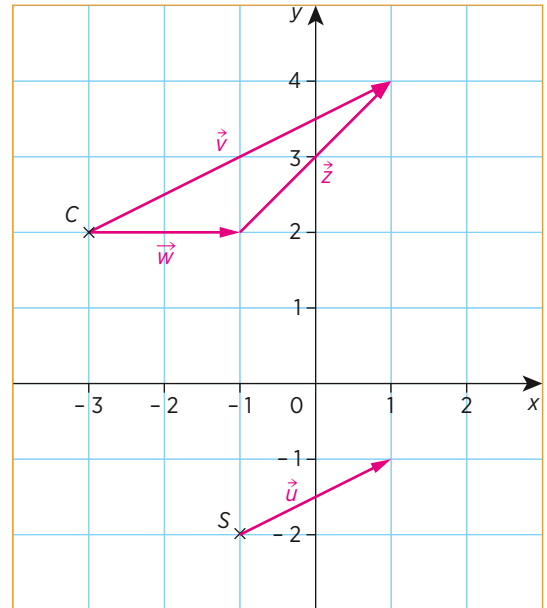


1 On appellera \vec{u} le vecteur traduisant le déplacement Est - Est - Nord et \vec{v} le vecteur résultant des deux translations consécutives de directions Est - Est puis Est - Est - Nord - Nord.

La Sardine Rose part du point $S(-1 ; -2)$
et *Le Calamar Doré* part du point $C(-3 ; 2)$.

a) Dans un repère orthonormé, tracer un représentant du vecteur \vec{u} d'origine S .

b) Tracer un représentant du vecteur \vec{v} d'origine C .



2 Compléter la phrase.

La Sardine Rose effectue une translation de vecteur \vec{u} de coordonnées $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ et *Le Calamar Doré* une translation de vecteur \vec{v} de coordonnées $\begin{pmatrix} 4 \\ 2 \end{pmatrix}$.

3 Compléter l'algorithme ci-dessous donnant les coordonnées du vecteur $\vec{v} \begin{pmatrix} a \\ b \end{pmatrix}$ résultant des translations successives de vecteurs $\vec{w} \begin{pmatrix} c \\ d \end{pmatrix}$ et $\vec{z} \begin{pmatrix} e \\ f \end{pmatrix}$ ($\vec{v} = \vec{w} + \vec{z}$).

Langage naturel

Variables	a, b, c, d, e, f nombres flottants.....
Entrée	Saisir c, d, e, f
Traitement	Affecter à a la valeur $c + d$ Affecter à b la valeur $e + f$
Sortie	Afficher « Les coordonnées du vecteur v sont : », a, b

4 Pour assurer la sécurité, les deux bateaux doivent avoir deux trajectoires parallèles. On suppose que les bateaux ont une trajectoire rectiligne.

Compléter l'algorithme ci-contre permettant de rentrer les coordonnées des vecteurs $\vec{u} \begin{pmatrix} x \\ y \end{pmatrix}$ et $\vec{v} \begin{pmatrix} a \\ b \end{pmatrix}$ associés respectivement à la trajectoire du bateau S et à celle du bateau C et d'indiquer en sortie si les bateaux naviguent en sécurité (trajectoires parallèles) ou non.

•••• RAPPEL

Deux vecteurs \vec{u} et \vec{v} sont colinéaires si et seulement si il existe un réel k tel que $\vec{v} = k\vec{u}$.

Langage naturel

Variables x, y, a, b, k nombres flottants
Entrée Saisir x, y, a, b
Traitement Si $a = 0$ et $b = 0$ alors
 Afficher « Bateaux en danger »
 Sinon
 Si $a = 0$ et $b \neq 0$ alors
 Afficher « Bateaux en danger »
 Sinon
 Si $a \neq 0$ et $b = 0$ alors
 Afficher « Bateaux en danger »
 Sinon
 Affecter à k la valeur $\frac{x}{a}$
 Si $y/b = k$ Alors
 Afficher « Bateaux en sécurité »
 Sinon
 Afficher « Bateaux en danger »
 Fin Si
 Fin Si
 Fin Si
 Fin Si

On doit traiter à part les cas où :
 $a = 0$ ou $b = 0$
 $a = 0$ et $b = 0$

Fin de l'algorithme

5 Écrire l'algorithme dans le langage de programmation Python, l'exécuter et indiquer si *La Sardine Rose* et *Le Calamar Doré* naviguent en sécurité.

```
x=float(input("x="))
y=float(input("y="))
a=float(input("a="))
b=float(input("b="))
if a==0 and b==0
    print(" Bateaux en danger ")
elif a==0 and b!= 0
    print(" Bateaux en danger ")
elif a!=0 and b==0
    print(" Bateaux en danger ")
else k=x/a
    if y/b=k
        print(" Bateaux en sécurité ")
    else
        print(" Bateaux en danger ")
Oui, ils naviguent en sécurité.
```

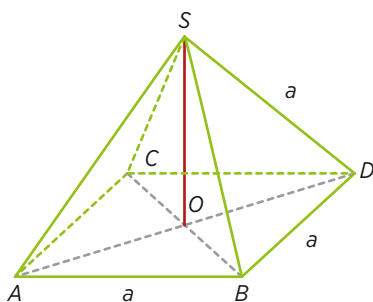
	Bateau 1	Bateau 2	Sécurité ou danger ?
a)	$\vec{u} \begin{pmatrix} 3 \\ 1 \end{pmatrix}$	$\vec{v} \begin{pmatrix} 6 \\ 3 \end{pmatrix}$	danger
b)	$\vec{u} \begin{pmatrix} -2 \\ 4 \end{pmatrix}$	$\vec{v} \begin{pmatrix} -1 \\ 3 \end{pmatrix}$	danger
c)	$\vec{u} \begin{pmatrix} -2 \\ -6 \end{pmatrix}$	$\vec{v} \begin{pmatrix} -1 \\ -3 \end{pmatrix}$	sécurité
d)	$\vec{u} \begin{pmatrix} 4 \\ -1 \end{pmatrix}$	$\vec{v} \begin{pmatrix} 16 \\ -4 \end{pmatrix}$	sécurité

6 Exécuter le programme pour les bateaux ci-contre et indiquer s'ils naviguent en sécurité ou non.

Exercice 1 Tracer des patrons

Boucle bornée

On considère une pyramide $SABCD$ de sommet S . Sa base est un carré $ABCD$ de côté a ($a > 0$). On note O le centre du carré. Les arêtes ont toutes pour longueur a .



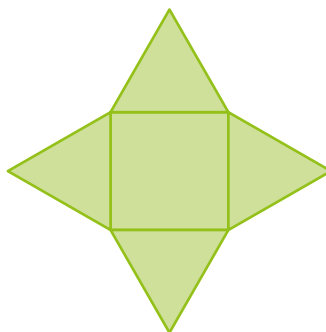
On considère le script ci-contre écrit avec le logiciel Scratch.

- 1 Quel est le rôle de ce script ? L'implanter sur un ordinateur et vérifier la réponse donnée.

Ce script permet le tracé d'un carré de centre $O(0, 0)$ et de côté a .



- 2 On admet que la figure ci-dessous représente un patron de la pyramide décrite ci-dessus.



- a) Quelles sont les mesures des côtés et les angles du triangle SAB ?

$$SA = SB = a$$

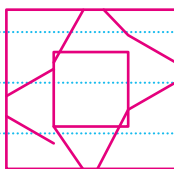
$$\widehat{SBA} = \widehat{SAB} = \widehat{ASB} = 60^\circ \text{ car le triangle } SAB \text{ est équilatéral.}$$

- b) En utilisant une boucle bornée, compléter la suite du script ci-contre pour qu'il trace le patron complet.

- c) Exécuter ce script avec $a = 50$ puis avec $a = 100$.

- d) Ce script répond-il au problème lorsque $a = 200$? Expliquer.

Pour $a = 200$, le carré central est tracé, mais le crayon atteint ensuite les bords de la fenêtre, ce qui le désoriente, on obtient :



Scratch

Répéter 4 fois

Tourner \curvearrowright de 60 degrés

Avancer de a

Tourner \curvearrowright de 120 degrés

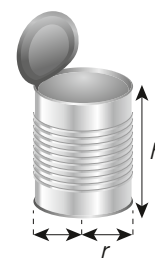
Avancer de a

Tourner \curvearrowright de 30 degrés

Exercice 2

Optimisation d'un coût de production

Notion de fonction



Les boîtes de conserve cylindriques sont fabriquées en découpant des tôles d'acier qui sont recourbées et ensuite soudées. Un industriel souhaite fabriquer des boîtes 4/4 d'une contenance de 850 mL (soit 850 cm³).

Le prix de l'acier étant élevé, il cherche les dimensions (hauteur h et rayon r , en mm) du cylindre qui répondra à la commande en utilisant la plus petite surface de tôle possible. On suppose ici que le rayon r appartient à l'intervalle $[0 ; 150]$.

1 Exprimer la hauteur h (en mm) d'une boîte répondant aux contraintes en fonction de son rayon r (exprimé en mm). $h = \frac{850\,000}{\pi r^2}$.

2 Compléter l'algorithme ci-contre de façon à ce qu'il affiche les hauteurs des cylindres possibles pour chaque valeur entière de rayon comprise entre 1 et 150 mm.

3 Donner l'expression de la surface $S(r, h)$ d'acier utilisée (exprimée en mm²) pour la fabrication d'une boîte de rayon r et de hauteur h (en mm) :

$$S(r, h) = 2 \times \pi \times r^2 + 2 \times \pi \times r \times h$$

4 Quel est le rôle du script ci-dessous ?

```
import math
def Surface(r,h):
    S=2*math.pi*r**2+2*math.pi*r*h
    return (S)
S=10**10
for r in range(1,151):
    h=850000/(math.pi*r**2)

    if Surface(r,h)<S:
        S=Surface(r,h)
        a=r
        b=h
print(a)
print(b)
```

5 Implanter ce programme sur un ordinateur.

a) Quelles sont les valeurs de a et b affichées en sortie ? $a = 51$ et $b \approx 104$.

b) Interpréter le résultat en revenant au contexte.

Si le rayon en mm est un nombre entier, c'est pour un rayon de 51 mm et une hauteur d'environ 104 mm que la surface d'acier utilisée pour fabriquer une boîte d'un volume de 850 mL est minimale.

Langage naturel

Variables	R, H nombres flottants
Traitement	Pour R allant de 1 à 150 Affecter à H la valeur $\frac{850\,000}{\pi R^2}$ Afficher H FinPour



BESOIN D'AIDE ?

La valeur initiale de S est choisie arbitrairement (ici 10^{10}) de façon à ce qu'elle soit largement supérieure aux différentes surfaces que l'algorithme va calculer ensuite.

Ce programme calcule les différentes surfaces pour toutes les valeurs entières de r comprises entre 0 et 150, et il stocke dans la variable S la surface minimale, dans a le rayon de la boîte correspondant et dans b la hauteur de boîte correspondante.

Exercice 3

L'éponge de Menger-Sierpinski

Boucle bornée

L'éponge de Menger est un objet mathématique que l'on fabrique en suivant la recette ci-dessous :

On prend un cube de 1 m d'arête.

Étape 1 : On découpe ce cube en 27 cubes identiques.

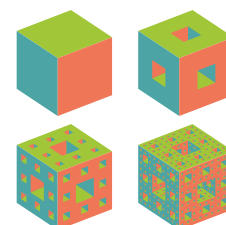
On enlève le cube central ainsi que les cubes situés au centre de chacune des faces.

Étape 2 : On répète l'étape 1 pour chacun des cubes restants.

Étape 3 : On répète l'étape 1 pour chacun des cubes restants après l'étape 2.

Étape 4 : On répète l'étape 1 pour chacun des cubes restants après l'étape 3.

Et ainsi de suite...



1 On applique l'étape 1 au cube initial d'arête 1 m.

a) Combien de cubes reste-t-il ? Il reste $27 - 7 = 20$ cubes.

b) Quelle est la mesure de leurs arêtes ? Chaque arête de cube mesure $\frac{1}{3}$ m.

c) Quel est le volume total du solide ainsi obtenu à cette étape ?

Le solide est composé de 20 cubes d'arête $\frac{1}{3}$ m donc $V = 20 \times \left(\frac{1}{3}\right)^3$.

2 On répète alors l'étape 1 à chacun des cubes restants.

a) Montrer que le nombre de cubes restants est 400 : $20 \times (27 - 7) = 400$.

b) Quelle est la mesure de leurs arêtes ?

Chaque arête mesure $\frac{1}{3} = \frac{1}{3^2}$ m.

c) Quel est le volume total du solide ainsi obtenu à cette étape ?

$V = 400 \times \frac{1}{3^2} \text{ m}^3$.

3 Compléter l'algorithme ci-contre qui, pour un entier N saisi, détermine et affiche à l'étape N :

- le nombre C de cubes restants
- la mesure A de leurs arêtes (en m)
- le volume V total du solide obtenu (en m^3)

4 a) À l'aide de la fiche calculatrice, traduire l'algorithme précédent dans le langage de programmation de votre calculatrice et implanter ce programme.

T.I. Basic

Prompt N

1 → C

1 → A

1 → V

For(I,1,N)

$20 \times C \rightarrow C$

$\frac{1}{3} \times A \rightarrow A$

$C \times A^3 \rightarrow V$

End

Disp.C,A,V

Basic Casio

? → N ↓

1 → C ↓

1 → A ↓

1 → V ↓

For 1 → I To N ↓

$20 \times C \rightarrow C$ ↓

$\frac{1}{3} \times A \rightarrow A$ ↓

$C \times A^3 \rightarrow V$ ↓

Next ↓

C ↓

A ↓

V ↓

Langage naturel

Variables

N, C, I nombres entiers

A, V nombres flottants

Entrées

Saisir N

Initialisation

C prend la valeur 1

A prend la valeur 1

V prend la valeur 1

Traitement

Pour I allant de 1 à N

C prend la valeur $C \times 20$

A prend la valeur $\frac{1}{3} \times A$

V prend la valeur $C \times A^3$

FinPour

Sortie

Afficher C

Afficher A

Afficher V

b) En déduire les valeurs de C, A et V à l'étape 20.

La calculatrice affiche : $C = 1,04 \times 10^{26}$ $A \approx 2,87 \cdot 10^{-10}$ (m) et $V \approx 0,0025$ (m^3).

c) Déterminer les valeurs de C, A et V à l'étape 50.

La calculatrice affiche : $C \approx 1,126 \cdot 10^{65}$ $A \approx 1,39 \cdot 10^{-24}$ (m) et $V \approx 3,04 \cdot 10^{-7}$ (m^3).

d) Comment semblent se comporter les variables C, A et V lorsque N prend des valeurs de plus en plus grandes ?

C prend des valeurs infiniment grandes, A et V semblent de plus en plus proches de 0.

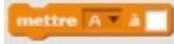


Création d'un nouveau programme

Fichier/Nouveau

Variables : saisie et affectation

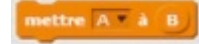
Saisir A



Afficher A



Affecter à A la valeur B



Tests et connecteurs logiques ET et OU

$A = B$



A et B



$A \geq B$



A ou B



Instruction conditionnelle : Si <condition> alors <instruction1> sinon <instruction2>

Si <condition> faire
<instruction1>
Sinon faire
<instruction2>
FinSi



Boucle bornée : Répéter a fois <instructions>

Pour i allant de a à b
<instructions>
FinPour



Boucle non bornée : Tant que <cond> faire <instructions>

Tant que <condition> faire
<instructions>
FinTantQue



Définir une fonction de paramètres a, b, c... qui retourne y

Définir une fonction de paramètres a, b et c via « Ajouter bloc »/
Créer un bloc/Options



Puissances et racine carrée

Puissances : on utilise le multiplicateur



\sqrt{x}



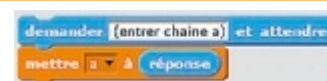
Nombres aléatoires

Nombre A entier aléatoire compris entre a et b



Chaînes de caractères

Saisir une chaîne **mot**



Longueur d'une chaîne **mot**



Extraire un caractère d'une chaîne de caractères **mot**



Concaténation de deux chaînes de caractères



Exécuter un programme

Dans Événements, utiliser un des blocs « quand... » et choisir l'instruction de départ.

Scratch est développé par the Lifelong Kindergarten Group au MIT Media Lab. Il est disponible gratuitement et téléchargeable sur <https://scratch.mit.edu>.



La pâte à papier utilisée pour la fabrication du papier de cet ouvrage provient de forêts certifiées et gérées durablement.

Imprimé en France par Pollina S.A. – 85400 Luçon
N° d'imprimeur : xxxx - Dépôt légal : avril 2018
N° d'édition : 03580204-01

Cahier d'algorithmique

Tout le programme en algorithmes

2^{de}

Belin:
ÉDUCATION

Avantages, offres spéciales, nouveautés, abonnez-vous
à la newsletter sur belin-education.com

belin-education.com

ISBN 979-10-358-0204-2



9 791035 802042

Code
03580204