L'algorithmique

1. Qu'est-ce qu'un algorithme ?

Un algorithme est une suite d'instructions élémentaires qui s'appliquent dans un ordre déterminé à des données et fournissant en un nombre fini d'étapes des résultats.
Une recette de cuisine est un algorithme. En effet il s'agit de :

- réunir les ingrédients ;
- préparer le plat ;
- arriver au plat prêt à être servi.

On peut considérer un algorithme comme une « boîte noire » fonctionnant selon le schéma suivant :

Exemple

Les données sont :	3 points distincts non alignés
Les instructions sont :	a) Tracer la médiatrice (D) de AB
	b) Tracer la médiatrice (D') de AC
	c) Appeler O le point d'intersection de (D) et (D')
	d) Tracer le cercle (C) de centre O et de rayon OA
Le résultat est :	le cercle circonscrit au triangle ABC

Exemple

Les programmes de calcul vu au collège sont des algorithmes. En effet, à partir d'un nombre on en « fabrique » un autre. Considérons l'algorithme « choisir n, lui ajouter 2, élever au carré le résultat et enfin ajouter 1 ».
Si en entrée on donne à n la valeur 3, quelle est la sortie ? . . . .

Bien sûr, il faudra procéder avec rigueur, sans ambigüité, pour obtenir un résultat. On va donc employer un langage spécifique.

2. Variables et affectations

Tout algorithme commence par le stockage des données qui seront utilisées lors du traitement. Chacune de ces données est stockée dans la mémoire de la calculatrice ou de l'ordinateur, à un emplacement nommé variable et repéré par un nom.
Les instructions que l'on peut pratiquer avec une variable sont les suivantes :

- **la saisie** : on demande à l'utilisateur de donner une valeur à la variable ;
- **l'affectation** : l'algorithme donne à la variable une valeur qui peut être le résultat d'une suite d'instructions ou bien une valeur venant d'être saisie par l'utilisateur. L'affectation est symbolisée par la flèche ← ;
- **l'affichage** : l'algorithme affiche la valeur de la variable.

3. Premiers exemples

Algorithme 1

Variables
V
Traitement
$V \leftarrow 4$
$V \leftarrow V + 7$
Sortie
Afficher V

Algorithme 2

Variables
a
Traitement
$a \leftarrow 1$
$a \leftarrow a + 3$
$a \leftarrow a^2$
$a \leftarrow a - 4$
Sortie
Afficher a

Algorithme 3

Variables
a, b, c
Traitement
$a \leftarrow 2$
$a \leftarrow a^2 - 1$
$b \leftarrow a + 1$
$c \leftarrow ab$
Sortie
Afficher c

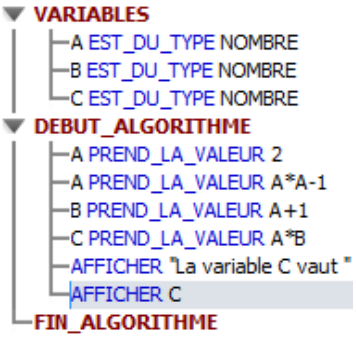
1. Écrire les valeurs prises successivement par la variable V de l'algorithme 1 : . . . . . Qu'affiche l'algorithme à la fin de son exécution ? . . .

	a	b	c
$a \leftarrow 2$			
$a \leftarrow a^2 - 1$			
$b \leftarrow a + 1$			
$c \leftarrow ab$			

2. Écrire les valeurs prises successivement par la variable a de l'algorithme 2 : . . . . . Qu'affiche l'algorithme à la fin de son exécution ? . . .

3. Utiliser le tableau ci-dessus pour donner les valeurs des variables a, b, c à chacune des étapes de l'algorithme 3.

4. Avec AlgoBox, l'algorithme 3 se programme comme montré sur l'impression d'écran.



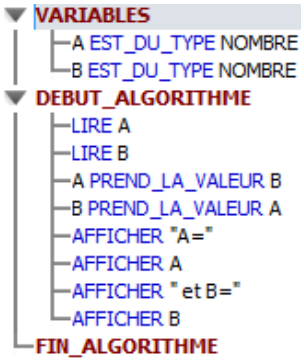
a. Écrire le programme et le tester.

b. Le modifier de façon à ce qu'à l'exécution du programme l'utilisateur soit invité à saisir la valeur de a qu'il désire.

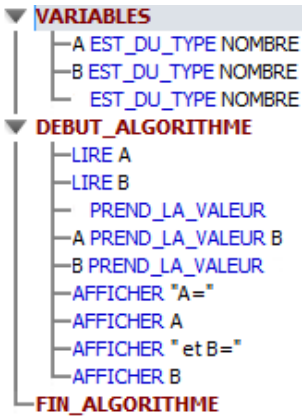
5. On souhaite écrire un algorithme qui permette d'échanger les valeurs de deux variables.

a. Programmer la première version de l'algorithme ci-contre et l'exécuter avec $A = 3$ et $B = 7$. Que retourne-t-il ? . . . . . Le résultat est-il satisfaisant ? . .

b. En utilisant une troisième variable, compléter la deuxième version de l'algorithme et la programmer.



Première version



Deuxième version

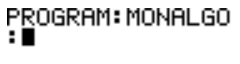
6. (Défi). On considère l'algorithme ci-contre à gauche. L'exécuter (à la main ou après l'avoir programmé) pour quelques valeurs de A. Quelle conjecture peut-on faire ? La démontrer.

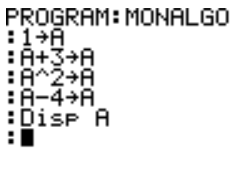
. . . . . .

Saisir A
$A \leftarrow A - 1$
$B \leftarrow A - 1$
$C \leftarrow B - 1$
$A \leftarrow A^2 + C$
$B \leftarrow A/B$
Afficher B

❖ **Programmation des exemples précédents**

Nous allons programmer l'exemple 2 sur la TI-82.
Appuyer sur le bouton prgm, puis choisir NOUV et enfin Nouveau. Saisir le nom du programme, par exemple MONALGO, puis valider. Le programme est prêt à être tapé.



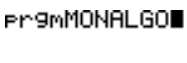


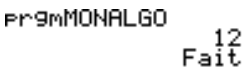
En langage TI, il est inutile de déclarer les variables, ce qui n'est pas le cas sur AlgoBox par exemple. La seule commande dont on a besoin pour ces divers algorithmes est Disp qui signifie « afficher ». **Cette commande n'est pas à taper en toute lettre.** Il faut aller la chercher dans le menu E/S (entrée/sortie) en appuyant sur prgm.

Manipulation pour obtenir certains caractères

Pour taper une lettre, appuyer au préalable sur alpha. Pour en taper plusieurs d'affilée, « verrouiller en alpha » en appuyant sur 2nde alpha. Pour sortir du mode alpha, appuyer une nouvelle fois sur alpha.
La flèche → s'obtient avec la touche sto→, les guillemets par alpha +, un espace par alpha 0, une virgule par , (oui !).

Pour exécuter le programme, il faut quitter l'éditeur de programme par la combinaison de touches 2nde mode (le programme s'enregistre automatiquement), puis appuyer à nouveau sur prgm et sélectionner cette fois-ci EXEC.
On choisit alors le programme par son numéro ou par les flèches haut/bas et on valide par entrer, ce qui a pour effet d'afficher la commande de lancement d'un programme (prgmEX2). Il faut alors valider une seconde fois pour lancer le programme. Le résultat s'affiche. Le message « Fait » signifie que l'exécution du programme est terminée.





7. Programmer l'algorithme 3 sur la calculatrice.

❖ **Amélioration de ces programmes**

Les algorithmes 1, 2 et 3 sont « figés », ils ne demandent rien à utilisateur. On aimerait modifier l'exemple 2 de façon à ce que lors de l'exécution du programme, une fenêtre de dialogue apparaisse et demande à l'utilisateur une valeur initiale de a. On remplace l'affectation « $a \leftarrow 1$ » par « Saisir a » (ou « Lire a » sur AlgoBox comme on l'a vu).

Il faut donc éditer (c'est-à-dire modifier) le programme existant. Appuyer sur prgm, puis choisir EDIT et MONALGO.

Algorithme 2 bis

Variables
a
Traitement
Saisir a
$a \leftarrow a + 3$
$a \leftarrow a^2$
$a \leftarrow a - 4$
Sortie
Afficher a

Pour supprimer une ligne, appuyer sur del (ou annul). La commande `Prompt` se trouve dans le même menu que `Disp`.
À présent lorsqu'on lance le programme, le nom de la variable apparaît, suivi d'un point d'interrogation. Il attend que l'on saisisse une valeur de a. Il est possible de relancer le programme immédiatement après l'avoir exécuté une première fois en appuyant sur entrer après « Fait ».

```
PROGRAM:MONALGO
:Prompt A
:A+3→A
:A^2→A
:A-4→A
:Disp A
:
```

```
prgmMONALGO
A=?1
                  12
                Fait
A=?2
                  21
                Fait
```

Apportons une ultime modification en embellissant le programme de quelques messages. Les commandes `Input` et `Disp` peuvent retourner un texte, à condition qu'il soit entre guillemets. Il faut séparer chaque message ou variable à retourner par des virgules.

```
PROGRAM:MONALGO
:Input "ENTREZ U
N NOMBRE",A
:A+3→A
:A^2→A
:A-4→A
:Disp "JE TROUVE
",A,"ET OUI !"
```

```
prgmMONALGO
ENTREZ UN NOMBRE
```

```
prgmMONALGO
ENTREZ UN NOMBRE
2
JE TROUVE
                  21
ET OUI !
                Fait
```

8. Écrire un algorithme qui étant donné les longueurs des deux côtés d'un triangle rectangle retourne la longueur de l'hypoténuse.

5. Instructions conditionnelles

❖ Présentation

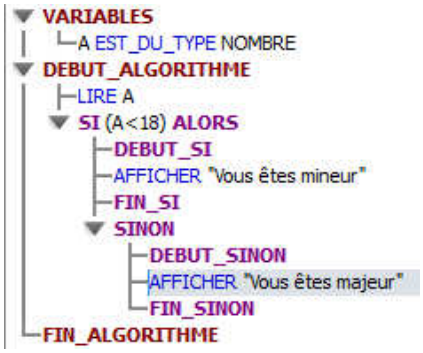
La résolution de certains problèmes nécessite la mise en place d'un test pour effectuer une tâche :

- si le test est positif, on effectue la tâche ;
- sinon, c'est-à-dire si le test est négatif, on effectue une autre tâche.

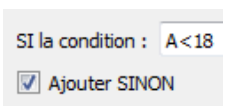
Le « sinon » n'est pas obligatoire. En son absence, lorsque le test est négatif, la tâche ne sera pas effectuée et l'algorithme passera à l'instruction suivante.

Le programme ci-contre demande l'âge d'une personne et affiche si elle est mineure ou majeure.

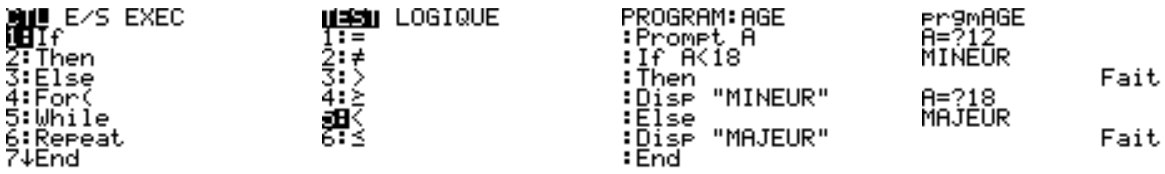
Traitement
Saisir a
Si $a < 18$ alors
    Afficher « vous êtes mineur »
    Sinon Afficher « vous êtes majeur »
FinSi



Il est aisé de traduire cet algorithme en AlgoBox, un peu de tâtonnement ne vous fera pas de mal. Seule précision : pour avoir le bloc SINON, il faut cocher la case « Ajouter SINON » lors de la création du bloc SI.



En TI, les commandes sont en anglais : `If` pour Si, `Then` pour Alors et `Else` pour Sinon. Elles se trouvent dans le menu `CTL` obtenu en appuyant sur prgm. Pour obtenir le symbole <, ouvrir le menu « test » par 2nde math.



❖ **Plusieurs conditions imbriquées**

On considère un algorithme qui affiche
- « normal » si la température est strictement supérieure à 5° ;
- « froid » si elle est comprise entre $-5°$ et 5° ;
- « grand froid » sinon.

On propose deux algorithmes, compléter le second et le programmer sur la calculatrice.

Traitement
Saisir T
Si $T > 5$
Alors Afficher « normal »
FinSi
Si $(T \geq -5)$ et $(T \leq 5)$
Alors Afficher « froid »
FinSi
Si $T < -5$
Alors afficher « grand froid »
FinSi

Traitement
Saisir T
Si …
Alors Afficher « normal »
Sinon Si …….
Alors afficher « froid »
Sinon afficher « grand froid »
FinSi
FinSi

9. Lors d'un excès de vitesse, la perte de point peut aller de 1 à 6 points selon la valeur du dépassement d de la vitesse autorisée.

d	$d < 20$	$20 \leq d < 30$	$30 \leq d < 40$	$40 \leq d < 50$	$d \geq 50$
Perte	1	2	3	4	6

Écrire un algorithme prenant en entrée le dépassement et retournant en sortie le nombre de points perdus.

10. Soit a et b deux réels. Résoudre l'équation $ax + b = 0$ (attention si $a = 0$!).
Écrire un algorithme demandant a et b et qui donne les solutions de cette équation.

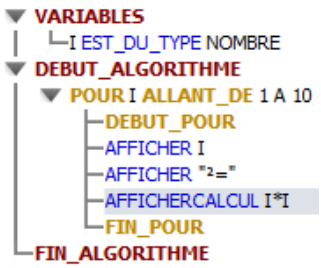
6. La boucle « pour »

Il est aussi possible de demander à un ordinateur de répéter une même tache autant de fois que l'on veut. Cela se fait grâce à ce qu'on appelle une **boucle**. Regardons de suite un exemple pour voir l'intérêt de cette notion.

Traitement
Pour K de 1 à 10 faire
Afficher K^2
Fin Pour

Le programme ci-contre affiche les carrés des nombres entiers de 1 à 10.
Le principe de fonctionnement est le suivant : l'ordinateur

affecte à K la valeur 1 comme il lui est demandé puis effectue les instructions comprises entre « Pour » et « Fin Pour ». Il augmente ensuite automatiquement K de 1 (donc K vaut 2) puis effectue à nouveau le bloc d'instruction. K est alors augmenté de 1 et ainsi de suite jusqu'à ce que K vaille 10.



La traduction en AlgoBox ne pose pas de difficultés particulières. Voir ci-contre une version embellie du programme.

```
PROGRAM:CARRE
:For(K,1,10)
:Disp K^2
:End
```

En Texas Instrument, la syntaxe est là encore en anglais :

`For(variable,début,fin)`

C'est avec l'introduction des boucles que l'intérêt de l'algorithmique en mathématiques se fait sentir. Voyons un exemple ci-dessous.

11. Que fait l'algorithme ci-contre ? Combien retourne-t-il ? . . . . . .
Généraliser en écrivant un programme qui calcule la somme $1+2+3+\cdots+n$ où n est demandé à l'utilisateur au début de l'exécution du programme.

Traitement
$S \leftarrow 0$
Pour K de 1 à 100 faire
  $S \leftarrow S+K$
Fin Pour
Afficher S

7. La boucle « tant que »

En algorithmique, on peut être amené à répéter un bloc d'instructions tant qu'une condition est vérifiée. En langage naturel, une telle répétition en boucle peut se formuler par « tant que », traduit par « while » dans la plupart des langages (sauf en AlgoBox).
Si la condition qui suit la ligne d'en-tête contenant l'instruction « tant que » est vérifiée, alors le programme exécute toutes les instructions du bloc qui suit ; à la fin de cette exécution il regarde à nouveau si la condition est vérifié et ainsi de suite jusqu'à ce qu'elle ne le soit plus.

Toute boucle « pour » peut être remplacée par une boucle « tant que » à l'aide d'un compteur. C'est ainsi que l'algorithme ci-contre affiche lui aussi les carrés des entiers de 1 à 100.
En revanche il existe des utilisations de « tant que » qui ne peuvent pas être remplacée par celle de « pour ».

Traitement
K prend la valeur 1
Tant que $K \leq 100$ faire
  Afficher K^2
  $K \leftarrow K+1$
Fin Tant que

12. L'effectif d'une population de bactéries double chaque jour. Si la population est de 100 bactéries le premier jour, au bout de combien de jours dépassera-t-elle 2000 bactéries ? (à la main) . . . . . .

Si la question avait été « au bout de combien de temps la population dépassera-t-elle 6 milliards de bactéries ? », le calcul à la main aurait été fastidieux.
On peut alors programmer l'algorithme ci-contre. J compte le nombre de jours et P est le nombre de bactéries le J-ième jour.

Traitement
J prend la valeur 0
P prend la valeur 100
Tant que $P \leq 6 \cdot 10^9$ faire
  $P \leftarrow 2 \times P$
  $J \leftarrow J+1$
Fin Tant Que
Afficher J

13. Une balle est lâchée de 100 mètres de haut et rebondit aux $4/5^{\text{ème}}$ de sa hauteur à chaque rebond. On considère que la belle ne rebondit plus si le rebond est inférieur à 1 cm. Au bout de combien de rebonds la balle ne rebondit plus ?