

1 Introduction

1.1 Les instructions

Un algorithme, c'est une succession d'instructions. Voici par exemple un algorithme très simple pour résoudre l'équation $6x + 7 = 0$:

- Soustraire 7 à chaque membre de l'équation
- Diviser chaque membre de l'équation par 6
- Afficher la solution de l'équation

1.2 La structure conditionnelle

On l'utilise lorsque l'on fait une étude de cas. C'est une structure du type suivant :

Structure d'algorithme 1. Structure conditionnelle

```

1  Si Condition, Alors
2      Instruction1
3      Instruction2
4      ...
5  Sinon
6      Instructiona
7      Instructionb
8      ...
9  Fin_Si

```

Exemple. Algorithme 1.1 — Préparer son cartable

```

1  Si on a cours de mathématiques, Alors
2      Prendre la calculatrice
3  Fin_Si

```

Exemple. Algorithme 1.2 — Signe du produit $a \times b$

```

1  Si Signe( $a$ ) == Signe( $b$ ), Alors
2      Signe(produit) = "+"
3  Sinon
4      Signe(produit) = "-"
5  Fin_Si

```

A faire. Algorithme 1.3 — Équation de la droite (AB)

```

1  Si  $x_A = x_B$ , Alors
2      L'équation est  $x = x_A$ 
3  Sinon
4      La pente est  $a = \frac{y_B - y_A}{x_B - x_A}$ 
5      L'ordonnée à l'origine est  $b = y_B - a \times x_B$ 1
6      L'équation est  $y = ax + b$ 
7  Fin_Si

```

1.3 La structure itérative

On l'utilise lorsque l'on doit répéter plusieurs fois une instruction. C'est une structure qu'on peut rencontrer sous plusieurs formes.

1. Ici, on a utilisé le point B mais bien sûr, cela fonctionne aussi avec le point A .

1.3.1 Forme “Pour chaque”

Structure d’algorithme 2.a — Structure itérative “Pour chaque”

```
1 Pour chaque élément dans un ensemble donné, Faire
2     Instruction1
3     Instruction2
4     ...
5 Fin_Pour_chaque
```

Exemple. Algorithme 2.a — Préparer son cartable

```
1 Pour chaque matière du lendemain, Faire
2     Prendre le cahier de la matière
3     Prendre le livre de la matière
4 Fin_Pour_chaque
```

1.3.2 Forme “Tant que”

Structure d’algorithme 2.b — Structure itérative “Tant que”


```
1 Tant que Condition, Faire
2     Instruction1
3     Instruction2
4     ...
5 Fin_Tant_que
```

Exemple. Algorithme 2.b — Comportement en classe

```
1 Tant que le professeur ou un camarade a la parole, Faire
2     Écouter la personne
3     Prendre en note sur son cahier
4 Fin_Tant_que
```

2 AlgoBox avec Proglab

Le logiciel AlgoBox permet de s’initier à la programmation. On peut télécharger le logiciel sur <https://www.xm1math.net/algoBox/>, ou bien l’utiliser en ligne sur <http://proglab.fr/>. Nous allons ici détailler le fonctionnement avec proglab :



The screenshot shows the Proglab web interface. At the top, there are navigation buttons: Ouvrir, Enregistrer, Partager, Copier, and Télécharger. Below this, there are tabs for Variables, Entrées et sorties, Boucles et tests, and Autres. The main area displays a code editor with the following code:

```
1 VARIABLES
2 DEBUT_ALGORITHME
3 FIN_ALGORITHME
```

Below the code editor, there is a sidebar with a list of categories: Variables, Calculs, Chaînes, Listes, Tests, Boucles, Affichages, Graphiques, F1, and F2. The right side of the interface shows a help page titled "AlgoBox - calculs". It contains an "Attention" note: "x^2 ne veut pas dire « x au carré ». Il faut utiliser pow(x, 2) pour cela." Below this is a table of mathematical operations and functions.

Calcul	Opérateur ou fonction
Les 4 opérations	+ - * /
x^2	pow(x, 2) ou x*x
x^n	pow(x, n)
Racine carrée	sqrt(x)
Fonctions trigonométriques	sin(x), cos(x), tan(x) (x en radians)
Exponentielle	exp(x)
Logarithme népérien	log(x)
Valeur absolue	abs(x)
Partie entière	floor(x)
Arrondi à l'entier le plus proche	round(x)

Le programme de départ comporte 3 lignes (voir capture d’écran).

a) Déclaration des variables

Il faut expliquer à l'ordinateur quelles variables vont servir. Pour cela, toute variable que l'on va utiliser doit être déclarée en cliquant sur `variable EST_DU_TYPE_NOMBRE` (on n'utilisera que les nombres pour l'instant). Une fois déclarée, elle apparaît en-dessous de la première ligne "VARIABLES"

b) Écriture du programme

Le programme doit être écrit entre "DEBUT_ALGORITHME" et "FIN_ALGORITHME". Pour ajouter une instruction au programme, il est nécessaire de créer d'abord une nouvelle ligne en cliquant à la fin d'une ligne de l'algorithme (donc, au départ, forcément à la fin de la ligne "DEBUT_ALGORITHME" et en appuyant sur "Entrée". La nouvelle ligne est insérée en-dessous de la ligne actuelle.

c) Exécution du programme

Cliquer sur le bouton "Play" : le programme s'exécute et ce qui doit s'afficher est à droite.

2.1 Notre premier programme : la moitié d'un nombre

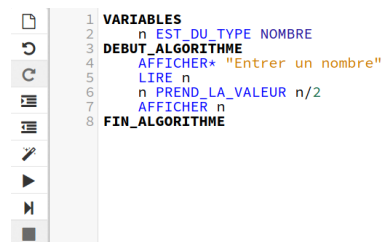
`variable EST_DU_TYPE_NOMBRE` et taper "n" (ce qui va remplacer le rond rouge).

`AFFICHER* "message"` et taper "Entrer un nombre" (ce qui va remplacer le rond orange).²

`LIRE variable` et taper "n".³

`variable PREND_LA_VALEUR valeur` et taper "n/2".

`AFFICHER* variable` et taper "n".



```
1 VARIABLES
2 n EST_DU_TYPE NOMBRE
3 DEBUT_ALGORITHME
4 AFFICHER* "Entrer un nombre"
5 LIRE n
6 n PREND_LA_VALEUR n/2
7 AFFICHER n
8 FIN_ALGORITHME
```

2.2 Tester si un nombre est pair

Effectuer un test, c'est utiliser une structure conditionnelle (Si... Alors...). On ajoute une structure conditionnelle avec `SI condition ALORS ...`

Lorsque l'on veut ajouter une instruction à faire dans le cas "Sinon", il faut à la place cliquer sur le `SI condition ALORS ... SINON ...`

Pour tester si un nombre est pair, on va utiliser la structure "Si... Alors..." et comme condition, on peut écrire le test " $n \% 2 == 0$ " (% désigne la fonction reste de la division entière). Un nombre est pair si et seulement si lorsqu'on le divise par 2, le reste est nul.

3 L'algorithme de Syracuse

L'algorithme de Syracuse suivant a été transcrit sous Algobox⁴. Testez-le!

Algorithme 3 — Suite de Syracuse

```
1 Choisir un nombre N
2 Tant que N > 1, Faire
3   Si N est pair, Alors
4     N = N/2
5   Sinon
6     N = 3 × N + 1
7   Fin Si
8   Afficher N
9 Fin_Tant_que
```

2. Ceci sert à afficher un message lors de l'exécution de l'algorithme. L'étoile est facultative mais permet que les affichages soient écrits à la ligne les uns des autres, ce qui est plus pratique.

3. Ceci demande à l'utilisateur de choisir un nombre lors de l'exécution, nombre affecté à la variable n.

4. <http://proglab.fr/txf848> ou <http://www.barsamian.am/S5P6/syracuse.alg> si vous voulez le fichier.

4 Histoires d'entrées et sorties

Algorithme de calcul.

Variables :

n et x sont deux nombres entiers.

Corps de l'algorithme :

```
1 Lire la variable  $n$ 
2 Si  $n < 10$ , Alors
3      $x$  prend la valeur  $n \times 3$ 
4 Sinon
5      $x$  prend la valeur  $n \times 3 + 6$ 
6 Fin_Si
7 Afficher la variable  $x$ 
```

1. Pour la valeur saisie $n = 5$, quel est le nombre affiché ?
2. Pour la valeur saisie $n = 13$, quel est le nombre affiché ?
3. Si le nombre affiché est 36, quelle était la valeur saisie ?

5 Promotions chez votre parfumeur

Dans une parfumerie on propose deux promotions différentes pour l'achat de deux articles :

- Formule n°1 : une réduction de 20% sur le montant total à payer.
- Formule n°2 : une réduction de 50% sur le prix de l'article le moins cher.

1. On s'intéresse dans cette question à la formule n°1.
 - (a) On achète chez le parfumeur un parfum affiché à 10€ et une lotion affichée à 20€. En bénéficiant de la réduction de la formule n°1, combien va-t-on payer ?
 - (b) Détailler la suite d'opérations qui vous a permis de partir des deux prix et de donner le montant final à payer.
 - (c) Programmer sous Algobox un algorithme qui affiche le montant de l'achat de deux articles selon la formule n°1 :
l'algorithme comportera les variables *article1* (prix de l'article 1), *article2* (prix de l'article 2) et *montant1* (montant à payer si on bénéficie de la formule n°1). Il pourra comporter d'autres variables si l'on a besoin de calculs intermédiaires.
on commencera par demander à l'utilisateur de rentrer les valeurs des deux articles, et on calculera dans *montant1* le montant de l'achat avant de l'afficher.
2. On s'intéresse maintenant à la formule n°2.
 - (a) Pour les mêmes achats que tout à l'heure, combien va-t-on payer si on bénéficie de la réduction de la formule n°2 ?
 - (b) Compléter le programme par le calcul et affichage du montant de l'achat de ces deux articles selon la formule n°2. On pourra utiliser une nouvelle variable *montant2*.
3. On veut maintenant comparer les deux formules.
 - (a) Terminer le programme en rajoutant l'affichage du nom de la formule la moins chère.
 - (b) A l'aide de ce programme, conjecturer quelle formule est la plus intéressante. . .
 - dans le cas particulier où les prix des deux articles sont identiques
 - dans le cas où le premier article coûte 20€ de plus que le second.
 - (c) Démontrer ces deux conjectures.