

# Algorithmique avec Algobox

## Fiche 1

### 1. Algorithme:

Un algorithme est une suite d'instructions qui, une fois exécutée correctement, conduit à un résultat donné  
Un algorithme doit contenir uniquement des instructions compréhensibles par celui qui devra l'exécuter.

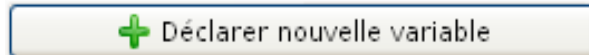
### 2. Algobox:

**Algobox** est un logiciel d'initiation à la programmation utilisant des instructions simples.

#### 2.1. Fonctionnement général

Le fonctionnement d'AlgoBox est basé sur les deux principes suivants :

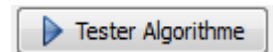
- Pour pouvoir utiliser une variable dans son algorithme, il faut d'abord la déclarer en utilisant le bouton qui est activable à tout moment.



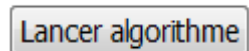
- Pour ajouter un nouvel élément à l'algorithme (autre que la déclaration d'une variable), il faut d'abord insérer une nouvelle ligne en utilisant le bouton.



- Une fois que l'on est positionné sur une nouvelle ligne, il suffit de cliquer sur l'un des boutons disponibles dans le panneau "Ajouter code" pour insérer l'instruction (ou le bloc d'instructions) désiré.
- Pour tester le fonctionnement de l'algorithme, il suffit de cliquer sur le bouton



- Une fenêtre apparaît alors dans laquelle il suffit de cliquer sur le bouton **Lancer algorithme** pour procéder à l'exécution de l'algorithme.



#### 2.2. Exemple : un ordinateur poli :

Écrire et exécuter l'algorithme suivant dans **Algobox**.

```
VARIABLES  
DEBUT_ALGORITHME  
    AFFICHER "Bonjour"  
FIN_ALGORITHME
```

#### 2.3. Une fonction affine :

Supposons que l'on veuille calculer l'image de n'importe quel réel  $x$  par la fonction  $f$  définie par  $f(x) = 3x + 4$ .

1. Déclarer la variable  $x$ .
2. Déclarer la variable  $y$ .
3. Ajouter une instruction permettant de lire la valeur de  $x$ .
4. Ajouter une instruction permettant de mettre  $3x + 4$  dans la variable  $y$ .
5. Ajouter une instruction permettant d'afficher la valeur de  $y$ .
6. Tester l'algorithme en remplissant le tableau de valeur:

$x$	-1	2	3	$\frac{1}{3}$
$f(x)$				

## 2.4. Une fonction à deux variables :

Concevoir un algorithme avec **Algobox** qui calcule l'aire d'un rectangle après que l'utilisateur ait entré la largeur et la longueur du rectangle. Utiliser trois variables qui porteront obligatoirement comme nom : largeur, longueur et aire.

## 3. Langage d'AlgoBox :

### 3.1. Les types de variables :

**Algobox** permet l'utilisation de trois types de variables :

- les nombres (type NOMBRE)
- les listes de nombres (type LISTE)
- les chaînes de caractères (type CHAINE).

### 3.2. Opérations avec les nombres :

En plus des 4 opérations de base (+,-,\*,/), les calculs possibles sont (  $x$  étant une variable du type NOMBRE) :

- Racine carrée d'une variable  $x$  : `sqrt(x)`
- Mettre une variable  $x$  à la puissance  $n$  : `pow(x,n)`
- Obtenir un nombre pseudo-aléatoire compris en 0 et 1 : `random()`
- Nombre  $\pi$  : `Math.PI`
- Partie entière d'une variable  $x$  : `floor(x)`
- Cosinus d'une variable  $x$  (en radians): `cos(x)`
- Sinus d'une variable  $x$  (en radians): `sin(x)`
- Tangente d'une variable  $x$  (en radians): `tan(x)`
- Valeur absolue d'une variable  $x$  : `abs(x)`
- Arrondi d'une variable  $x$  à l'entier le plus proche : `round(x)`
- Reste de la division de la variable  $x$  par la variable  $y$  : `x%y`

### 3.3. Copier, coller et couper certaines lignes de l'algorithme :

Via le menu Édition, il est possible de copier/coller/couper une ligne de code, mais cette option n'est valable que pour les lignes du type ...PREND\_LA\_VALEUR..., AFFICHER.... et TRACER.....

Remarque : pour coller une ligne de code, il faut d'abord créer une nouvelle ligne conformément aux règles d'édition en vigueur sur Algobox.

### 3.4. Tracer des points et des segments dans un repère :

En activant l'option Utiliser un repère dans l'onglet "Dessiner dans un repère", un repère graphique est automatiquement ajouté dans la fenêtre de test de l'algorithme. Il est alors possible d'inclure dans le code de l'algorithme des instructions pour tracer des points et des segments dans ce repère en utilisant les boutons « Ajouter TRACER POINT » et « Ajouter TRACER SEGMENT » .

### 3.5. Premières règles concernant la conception d'un algorithme avec Algobox :

- Toujours commencer par déterminer et déclarer les variables nécessaires à la bonne marche de l'algorithme. Pour cela, utiliser le bouton « *Déclarer nouvelle variable* ». Pour les noms de variables, ne pas oublier que votre algorithme va être lu par diverses personnes : donner des noms compréhensibles, sans ambiguïté, mais pas trop longs non plus !
- Pour ajouter un nouvel élément à l'algorithme (autre que la déclaration d'une variable), il faut d'abord insérer une nouvelle ligne en se positionnant à l'endroit adéquat et en cliquant sur le bouton « *Nouvelle Ligne* ».
- La déclaration d'une variable est comme une réservation d'une place de parking, vide au début. Pour que l'utilisateur puisse y mettre des données, il faut utiliser le bouton « *Ajouter LIRE variable* ».
- Pour donner une valeur à une variable (après un calcul éventuellement) à l'intérieur de l'algorithme, il faut utiliser le bouton « *AFFECTER valeur à variable* ». La boîte de dialogue qui apparaît permet de sélectionner la valeur à laquelle on veut affecter une valeur et l'expression (ou le calcul) qui permet d'obtenir cette valeur.
- Pour pouvoir afficher un résultat correspondant à la valeur d'une variable, il faut utiliser le bouton « *Ajouter AFFICHER variable* » et sélectionner la variable en question dans la boîte de dialogue qui apparaît.

### 3.6. Exemples de syntaxe pour Algobox :

- Pour vérifier si x est égal à 2, la condition à écrire est :  $x==2$
- Pour vérifier si x est différent de 2, la condition à écrire est :  $x!=2$
- Pour vérifier si x est strictement inférieur à 2, la condition à écrire est :  $x<2$
- Pour vérifier si x est inférieur ou égal à 2, la condition à écrire est :  $x<=2$
- Pour vérifier si x est strictement supérieur à 2, la condition à écrire est :  $x>2$
- Pour vérifier si x est supérieur ou égal à 2, la condition à écrire est :  $x>=2$

## 4. Premiers algorithmes :

### 4.1. Distance de sécurité :

Par sécurité, un véhicule doit respecter une distance minimale avec le véhicule qui le précède, afin d'avoir le temps de freiner avant une collision. Ce temps correspond à celui de la perception puis de la réaction du conducteur, ainsi que des possibilités de freinage du véhicule.

Ce temps est fonction de la vitesse du véhicule. Des études statistiques ont montré que cette distance peut être calculée par la formule :  $D = 8 + 0,2 v + 0,003 v^2$ , où  $v$  est en  $kmh^{-1}$  et  $D$  en mètres.

Créer un algorithme (dans le tableau ci-dessous) qui nous donne cette distance en fonction de la vitesse.

Votre algorithme	Vos résultats				
	Vitesse	50	90	110,5	130
	Distance de sécurité				

### 4.2. Distance entre deux points :

Créer un algorithme qui donne la distance entre deux points du plan connus par leurs coordonnées.

Les résultats seront donnés dans le tableau ci-dessous à  $10^{-2}$  près.

Votre algorithme	Vos résultats				
	Abscisse de A	3	- 10	$\sqrt{3}$	18 452
	Ordonnée de A	5	4	$\frac{22}{7}$	32 568
	Abscisse de B	8	- 7	18	- 795
	Ordonnée de B	- 2	13	185	9 501
	Distance AB à $10^{-2}$ près				

### 4.3. Coordonnées du milieu d'un segment dans un repère :

#### 4.3.1. Étude analytique :

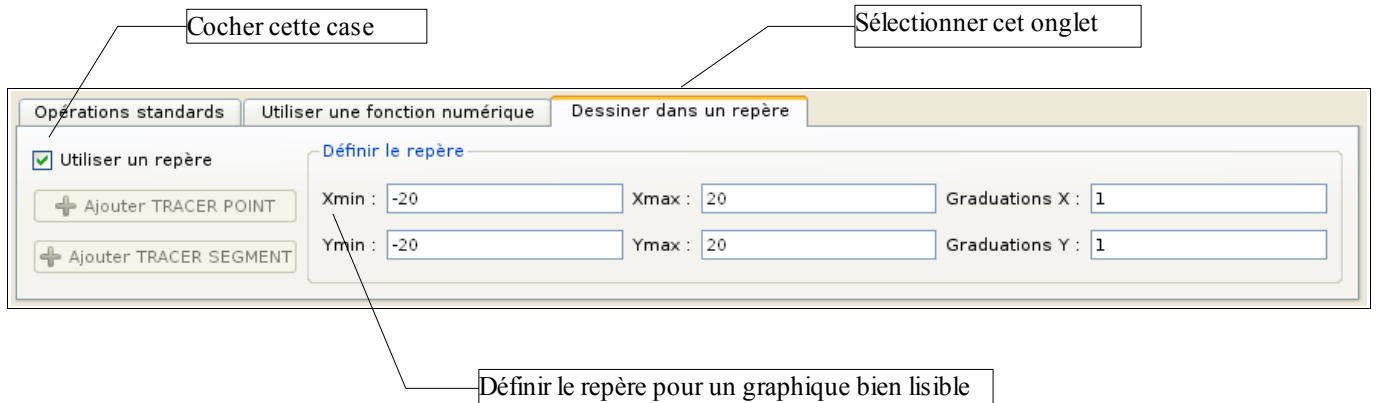
Créer un algorithme qui donne les coordonnées du milieu de deux points du plan connus par leurs coordonnées.

Les résultats seront donnés dans le tableau ci-dessous à  $10^{-2}$  près.

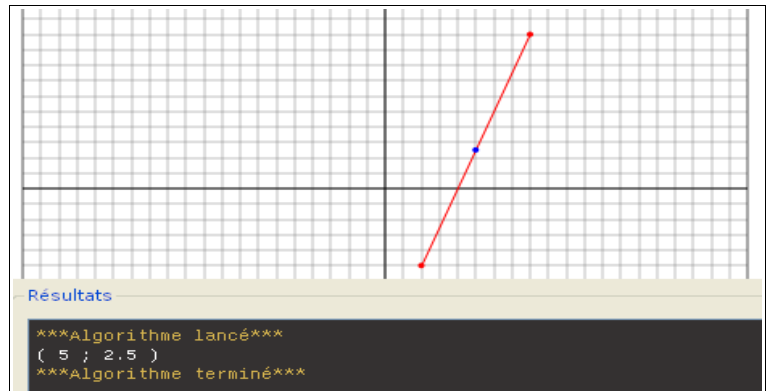
A	(2 ; -5)	(-7 ; -7)	(3 ; 5)
B	(8 ; 10)	(3 ; $\sqrt{2}$ )	( $\pi$ ; 5)
I			

#### 4.3.2. Illustration géométrique :

Conjointement à l'algorithme et au programme associé, on peut faire représenter la situation dans un repère.



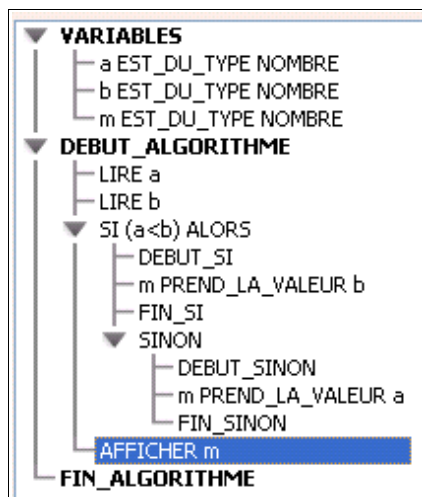
Voici un exemple de résultat attendu pour le premier exercice qui vous est proposé.



### 5. Instructions conditionnelles :

#### 5.1. Lecture d'un algorithme :

**ORDINATEUR INTERDIT** : Après d'éventuels essais papier crayon, indiquer ce que fait cet algorithme.



Après avoir lu les données a et b, cet algorithme affiche m qui représente : .....

## 5.2. Indice de masse corporelle :

Cette notion est apparue au XIX<sup>e</sup> siècle. Il s'en est suivie celle d'Indice de Masse Corporelle (ou IMC en français et BMI en anglais pour Body Mass Index). Elle permet de connaître quel serait son « poids idéal » à partir d'une formule.

Pour tenir compte des réalités naturelles des différences de constitution entre femmes et hommes, le scientifique H. A. Lorentz a établi les formules suivantes :

*Le poids idéal pour une femme serait égal à :*  $Taille(cm) - 100 - [Taille(cm) - 150]/2$   
*Le poids idéal pour un homme serait égal à :*  $Taille(cm) - 100 - [Taille(cm) - 150]/4$

*Avec comme conditions d'utilisation de ces formules :*

- *âge supérieur à 18 ans*
- *taille entre 140 et 220 cm*

Activité : créer, avec Algobox, un programme nommé IMCFH qui demande le sexe de la personne et qui, suivant la réponse, donnera ce que serait le poids idéal.

ATTENTION: pour la variable « Sexe », vous avez le choix entre une variable de type « Nombre » ou de type « Chaîne de caractères ». Dans les deux cas, faites porter le test uniquement sur un caractère : Femme par exemple.

Pour la variable de type « Nombre » Femme → 2

Pour la variable de type « Chaîne » Femme → F Dans la syntaxe de l'algorithme, le contenu d'une chaîne doit être encadré par des guillemets : Exemple : *a prend la valeur "bonjour"* (a étant une variable du type chaîne)

SI (Sexe=="F") ALORS

N.B. Algobox fait la différence entre majuscules et minuscules ! ! ! !

Votre algorithme	Vos résultats															
De quelles variables a-t-on besoin ? (indiquez leurs noms) :	Pour contrôler votre programme, Complétez le tableau suivant qui sera vérifié par votre professeur :															
Sur quelle variable doit-on appliquer la condition SI ?	<table border="1"> <thead> <tr> <th></th> <th>150 cm</th> <th>160 cm</th> <th>180 cm</th> <th>2m 10</th> </tr> </thead> <tbody> <tr> <td>Femme</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Homme</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		150 cm	160 cm	180 cm	2m 10	Femme					Homme				
	150 cm	160 cm	180 cm	2m 10												
Femme																
Homme																
L'algorithme :	<p>Pour les esprits curieux et critiques :</p> <p>Quel poids idéal pour une femme de 80 cm ?</p> <p>Quel poids idéal pour un homme de 80 cm ?</p> <p>Pourquoi ces résultats ?</p>															

# Fiche professeur

## Résultats attendus :

Fonction affine  
Page 1 § 2.3.

$x$	-1	2	3	$\frac{1}{3}$
$f(x)$	1	10	13	5

Distance de sécurité  
Page 3 § 4.1.

Vitesse	50	90	110,5	130
Distance de sécurité	25,5	50,3	66,73	84,7

Distance entre deux points  
Page 4 § 4.2.

Abscisse de A .....	3	- 10	$\sqrt{3}$	18 452
Distance AB à $10^{-2}$ près	8,6	9,49	182,58	30042,2

Coordonnées du milieu  
Page 4 § 4.3.1.

A .....	(2 ; -5)	(-7 ; -7)	(3 ; 5)
I	(5;2,5)	(-2;-2,79)	(3,07;5)

I M C  
Page 6 § 5.2.

	150 cm	160 cm	180 cm	2m 10
Femme	50	55	65	80
Homme	50	57,5	72,5	95