



Chapitre XII - Algorithmique

Bacomathiques — <https://bacomathiqu.es>

Table des matières

I - Définition	1
II - Instructions	1
1. Création de variables	1
2. Affections de valeurs	2
3. Affichage de variables	3
III - Blocs d'instructions	4
1. Définition	4
2. Les blocs SI et SINON	4
3. La boucle POUR	5
4. La boucle TANT QUE	6
IV - Algorithmes sur l'ordinateur	6

I - Définition

Un **algorithme** est une suite finie et ordonnée d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat. Ainsi, faire une recette de cuisine ou encore effectuer une division euclidienne à la main sont des exemples d'algorithmes.

Nous ne travaillerons dans ce cours qu'avec des algorithmes "type BAC".

II - Instructions

1. Création de variables

Créer une **variable** permet de réserver un espace pour y stocker des données quelconques. On donne un nom à chaque espace pour le repérer : ce sont les noms de variables. On leur donne également un type (entier, réel, ...) pour travailler avec.

Exemple (tiré du sujet de Pondichéry 2017) :

Variables	R et S sont des réels n et k sont des entiers
Traitement	S prend la valeur 0 Demander la valeur de n Pour k variant de 1 à n faire R prend la valeur $\frac{2,5}{n} \times f\left(\frac{2,5}{n} \times k\right)$ S prend la valeur $S + R$ Fin Pour Afficher S

Ici nous avons quatre variables : R et S qui sont des réels et n et k qui sont des entiers.

2. Affections de valeurs

Comme dit précédemment, les variables sont des “espaces” dans lequel il est possible de stocker des informations.

Cependant, après avoir créé cet espace, celui-ci est encore vide. C’est pourquoi on doit le “remplir” : c’est **l’affectation d’une valeur à une variable**.

Il existe plusieurs manières d’affecter une valeur à une variable : soit on lui donne directement sa valeur dans l’algorithme, soit on demande à l’utilisateur d’entrer une valeur (il faut garder à l’esprit que les algorithmes sont faits pour être utilisés par des utilisateurs).

Exemple (tiré du sujet de Pondichéry 2017) :

Variables	R et S sont des réels n et k sont des entiers
Traitement	S prend la valeur 0 Demander la valeur de n Pour k variant de 1 à n faire R prend la valeur $\frac{2,5}{n} \times f\left(\frac{2,5}{n} \times k\right)$ S prend la valeur $S + R$ Fin Pour Afficher S

Ici on donne à S la valeur 0, mais on demande à l’utilisateur d’entrer la valeur de la variable n (l’utilisateur entrera un entier, car la variable n ne peut contenir que des entiers).

Une fois que l’on a affecté une valeur à une variable, il est encore possible de la changer !

3. Affichage de variables

Les algorithmes étant faits pour être utilisés, il faut donc **retourner un résultat** sinon ceci seraient inutiles.

C'est pourquoi, on peut "afficher" les valeurs des variables (les montrer à l'utilisateur).

Exemple (tiré du sujet de Métropole 2017) :

Variables	N et A des entiers naturels
Entrée	Saisir la valeur de A
Traitement	N prend la valeur 0 Tant que $N - \ln(N^2 + 1) < A$ N prend la valeur $N + 1$ Fin tant que
Sortie	Afficher N

Une fois l'algorithme terminé, on affiche la valeur de la variable N (on remarque que N a pris plusieurs valeurs différentes au cours de l'algorithme mais qu'on affiche uniquement la valeur finale de la variable).

III - Blocs d'instructions

1. Définition

Les blocs d'instructions sont des parties de l'algorithme (ce sont des "algorithmes dans l'algorithme") qui s'exécutent suivant certaines conditions propres aux différents blocs d'instructions.

2. Les blocs SI et SINON

Les blocs **SI** et **SINON** sont des blocs d'instructions très utilisés qui permettent de tester une condition : si elle est réalisée, on va exécuter les instructions se situant sous le bloc SI et sinon, on va exécuter celles se situant sous le bloc SINON.

Exemple (test de parité) :

Variables	N et R sont des entiers
Entrée	Saisir la valeur de N
Traitement	SI $E(N/2) = N/2$: $R = 0$ SINON : $R = 1$
Sortie	Afficher R

Si la partie entière de $\frac{N}{2}$ est égale à $\frac{N}{2}$ (ce n'est vrai que pour les entiers pairs), alors on donne à R la valeur 0. Sinon on lui donne la valeur 1.

En fin d'algorithme, on affiche la valeur de R : soit 0 si N est pair, soit 1 si N est impair.

3. La boucle POUR

La boucle **POUR** est un bloc d'instruction qui s'exécute et qui va prendre à une variable toutes les valeurs comprises dans un ensemble d'entiers.

Exemple (calcul des termes d'une suite) :

Variables	U est un réel N et n sont des entiers
Entrée	Saisir la valeur de N
Traitement	Affecter à u la valeur 1 Pour n allant de 1 à N : Affecter à u la valeur $u + 1/n$
Sortie	Afficher n Afficher u

Cet algorithme permet de calculer les termes d'une suite définie par récurrence u_n :

$$\begin{cases} u_0 = 1 \\ u_{n+1} = u_n + \frac{1}{n} \end{cases} \quad \text{pour } n \text{ entier.}$$

On demande à l'utilisateur d'entrer une variable N , et pour n variant de 1 jusqu'à N (n prendra tour à tour les valeurs 1, 2, 3, ..., $N - 1$, N), on va calculer les termes de la suite.

4. La boucle TANT QUE

Cette boucle, différente de la boucle POUR permet d'exécuter son bloc d'instructions tant qu'une certaine condition est valable.

Exemple (tiré du sujet de Métropole 2017) :

Variables	N et A des entiers naturels
Entrée	Saisir la valeur de A
Traitement	N prend la valeur 0 Tant que $N - \ln(N^2 + 1) < A$ N prend la valeur $N + 1$ Fin tant que
Sortie	Afficher N

Ici, tant que $N - \ln(N^2 + 1)$ est inférieure à A , on affecte une nouvelle valeur à la variable N .

IV - Algorithmes sur l'ordinateur

Il est possible de tester et de vous entraîner aux algorithmes sur votre ordinateur, voir directement sur votre smartphone !

Divers logiciels à télécharger, dont certains ne nécessitent pas d'installation sont disponibles :

- Algogo
- AlgoBox
- Scratch
- Proglab
- PluriAlgo