

**Ministère de L'Éducation
Direction Générale de l'Enseignement
Préparatoire et Secondaire**



**CONVENTIONS D'ENSEIGNEMENT
DE L'ALGORITHMIQUE**

**2^{ème} Année Sciences
2^{ème} Année TI**

Septembre 2019

A. L'algorithme

1. Introduction

Dans le but de développer le raisonnement et la capacité de résolution des problèmes chez l'apprenant, le domaine « Pensée computationnelle et programmation » met l'accent sur l'algorithmique.

La partie analyse qui précède l'élaboration de l'algorithme, peut prendre différentes formes (organigramme, schéma, carte heuristique, ...).

2. Forme générale

ALGORITHME *Nom*

DEBUT

Traitements

FIN

Déclaration des objets

Objet	Type / Nature

B. Les syntaxes des structures algorithmiques

1. Les opérations élémentaires simples

a. L'opération d'entrée

Lire (Objet)

b. L'opération de sortie

Afficher ("Message", Objet, Expression)

c. L'opération d'affectation

Objet ← Expression

2. Les structures de données

• Les types de données

- Entier
- Réel
- Booléen
- Caractère
- Chaîne
- Tableau (Tableau de Taille_tableau de Type _élément)

3. Les structures de contrôle conditionnelles

• La structure de contrôle conditionnelle simple

Initialisation

Si Condition **Alors**

Traitement1

Sinon

Traitement2

FinSi

- *La structure de contrôle conditionnelle généralisée*

Initialisation

Si Condition 1 **Alors** Traitement 1

Sinon Si Condition 2 **Alors** Traitement 2

Sinon Si Condition 3 **Alors** Traitement 3

Sinon

.....

.....

Sinon Si condition N-1 **Alors** Traitement N-1

Sinon Traitement N

FinSi

- *On pourrait utiliser la structure de contrôle conditionnelle à choix multiples*

Initialisation

Selon <identificateur>

valeur(s) : instruction(s)

valeur(s) : instruction(s)

.....

[Autres : instructions]

Fin Selon

4. Les structures de contrôle itératives

- a. *La structure de contrôle itérative complète*

Initialisation

Pour Compteur de Début à Fin (Pas) Faire

Traitement

Fin Pour

b. La structure de contrôle itérative à condition d'arrêt

- Première formulation

Initialisation

Tant que condition(s) **Faire**

Traitement

Fin Tant que

- Deuxième formulation

Initialisation

Répéter

Traitement

Jusqu'à condition(s)

5. Les modules

a. Les Fonctions

- L'appel

Objet ← Nom_fonction (pe₁, pe₂, ... , pe_n)

- L'entête de la définition

Fonction Nom_fonction (pf₁: type₁, pf₂: type₂, ... , pf_n: type_n) : type_fonction

b. Les Procédures

- L'appel

Nom_procédure (pe₁, pe₂, ... , pe_n)

- L'entête de la définition

Procédure Nom_procédure (pf₁: type₁, pf₂: type₂, ... , pf_n: type_n)

- Mode de passage

Si le mode de passage est par variable (ou par adresse), on ajoutera le terme

Var avant le nom du paramètre.

Remarques :

- Pour les opérateurs arithmétiques et logiques, on va se baser sur ceux qui sont standards et développées dans le tableau suivant :

Opérateurs arithmétiques	
Opération	Opérateur
Somme	+
Soustraction	-
Multiplication	*
Division	/
Division entière	Div
Reste de la division entière	Mod

Opérateurs de comparaison	
Opération	Opérateur
Egal	=
Différent	≠
Strictement supérieur	>
Supérieur ou égal	≥
Strictement inférieur	<
Inférieur ou égal	≤
Appartient	∈

Opérateurs Logiques	
Négation	Non
Conjonction	Et
Disjonction	Ou
Disjonction exclusive	Ouex

- Pour les fonctions arithmétiques standards, on va se baser sur celles qui sont standards et développées dans le tableau suivant :

Algorithmique	Rôle
Arrondi(d)	Retourne un nombre qui est la valeur de d arrondie à la plus proche valeur.
Racine Carré(d)	Retourne la racine carrée de d si d est positif, sinon elle provoque une erreur.
Aléa(vi, vf)	Retourne un entier aléatoire de l'intervalle [vi, vf].

- Pour les fonctions sur les caractères, on va se baser sur celles qui sont standards et développées dans le tableau suivant :

Algorithmique	Rôle
Ord(c)	Retourne le code ASCII du caractère c .
Chr(d)	Retourne le caractère dont le code ASCII est d .

- Pour les fonctions sur les chaînes de caractères, on va se baser sur celles qui sont standards et développées dans le tableau suivant :

Algorithmique	Rôle
+	Permet la concaténation d'un ensemble de chaînes de caractères.
Long (ch)	Retourne le nombre de caractères de la chaîne ch .
Pos (ch1, ch2)	Retourne la première position de la chaîne ch1 dans la chaîne ch2 .
Convch (d)	Retourne la conversion d'un nombre d en une chaîne de caractères.
Valeur (ch)	Retourne la conversion d'une chaîne ch en une valeur numérique, sinon elle provoque une erreur.