

Ministère de l'Éducation
Direction Générale des Programmes
et de la Formation Continue



LES CONVENTIONS ALGORITHMIQUES

Septembre 2020

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. L'écriture d'un algorithme doit respecter les conventions citées dans ce document.

A. La forme générale d'un algorithme

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

Ecrire ("Message", Objet, Expression)

c. L'opération d'affectation

Objet ← Expression

2. Les types de données simples

- Entier
- Réel
- Booléen
- Caractère
- Chaîne de caractères

3. Les structures de données

- Tableau (à une ou à deux dimensions)
- Enregistrement
- Fichier

4. Les déclarations

a. Les objets de type de donnée simple

Objet	Type / Nature
Nom_objet	Type_objet

b. Les tableaux

	Objet	Type / Nature
Tableau à une dimension	Nom_tableau	Tableau de N Type _élément
Tableau à deux dimensions	Nom_tableau	Tableau de N lignes * M colonnes Type _élément

c. L'enregistrement

Objet	Type / Nature
Nom_enregistrement	Enregistrement Nom_champ1 : Type_champ1 Nom_champ2 : Type_champ2 ... Fin

d. Les fichiers

	Objet	Type / Nature
Fichier texte	Nom_fichier	Fichier Texte
Fichier de données	Nom_fichier	Fichier de Type _élément

e. Les nouveaux types utilisateurs

	Nouveau type
Tableau à une dimension	Nom_type = Tableau de N Type _élément
Tableau à deux dimensions	Nom_type = Tableau de N lignes * M colonnes Type _élément
Enregistrement	Nom_type = Enregistrement Nom_champ1 : Type_champ1 Nom_champ2 : Type_champ2 ... Fin
Fichier de données	Nom_type = Fichier de Type _élément

5. Les structures de contrôle conditionnelles

a. La structure de contrôle conditionnelle simple

Si Condition **Alors**

Traitement

FinSi

b. La structure de contrôle conditionnelle complète

Si Condition **Alors**

Traitement1

Sinon

Traitement2

FinSi

c. La structure de contrôle conditionnelle généralisée

Si Condition1 **Alors**

Traitement1

Sinon Si Condition2 **Alors**

Traitement2

Sinon Si Condition3 **Alors**

Traitement3

Sinon

Sinon Si conditionN-1 **Alors**

TraitementN-1

[**Sinon**

TraitementN]

FinSi

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

Selon <Expression>

Valeur1_1, Valeur1_2, ... : instruction(s)

Valeur2_1, Valeur2_2, ... : instruction(s)

.....

[**Autres** : instructions]

Fin Selon

6. Les structures de contrôle itératives

a. La structure de contrôle itérative complète

Pour Compteur de Début à Fin [Pas = valeur_pas] **Faire**

Traitement

Fin Pour

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

- Première formulation

Tant que condition **Faire**

Traitement

Fin Tant que

- Deuxième formulation

Répéter

Traitement

Jusqu'à condition

7. Les modules

a. Les Fonctions

- La déclaration

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

DEBUT

Traitement

Retourner Résultat

FIN

- L'appel

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

b. Les Procédures

- La déclaration

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

DEBUT

Traitement

FIN

- L'appel

Nom_procédure (pe₁, pe₂, ... , pe_n)

- Le mode de passage

Si le mode de passage est par référence (par adresse), on ajoutera le symbole @ avant le nom du paramètre.

8. Les opérateurs arithmétiques et logiques

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

Opérateurs Logiques	
Négation	Non
Conjonction	Et
Disjonction	Ou
Disjonction exclusive	Ouex

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

9. Les fonctions prédéfinies

a. Les fonctions sur les types numériques

Algorithmique	Rôle
Arrondi (x)	Retourne l'entier le plus proche de la valeur de x.
RacineCarré (x)	Retourne la racine carrée d'un nombre x positif.
Aléa (vi, vf)	Retourne un entier aléatoire de l'intervalle [vi, vf].
Ent (x)	Retourne la partie entière de x.

b. Les fonctions sur le type caractère

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

c. Les fonctions sur le type chaînes de caractères

Algorithmique	Rôle
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 (x)	Retourne la conversion d'un nombre x en une chaîne de caractères.
Estnum (ch)	Retourne Vrai si la chaîne ch est convertible en une valeur numérique, elle retourne Faux sinon.
Valeur (ch)	Retourne la conversion d'une chaîne ch en une valeur numérique, si c'est possible.
Sous_chaine (ch, d, f)	Retourne une partie de la chaîne ch à partir de la position d jusqu'à la position f (f exclue).
Effacer (ch, d, f)	Efface des caractères de la chaîne ch à partir de la position d jusqu'à la position f (f exclue).
Majus (ch)	Convertit la chaîne ch en majuscules.

10. Les fonctions et les procédures prédéfinies sur les fichiers

a. Les fichiers de données

Algorithmique	Rôle
Ouvrir ("Chemin\Nom_physique", Nom_logique, "Mode")	Ouverture d'un fichier <ul style="list-style-type: none">• Mode d'ouverture :<ul style="list-style-type: none">○ "rb" : Lecture (pointer au début)○ "wb" : Ecriture (création)○ "ab" : Ecriture à la fin du fichier
Lire (Nom_logique, Objet)	Lecture d'un objet à partir d'un fichier
Ecrire (Nom_logique, Objet)	Ecriture d'un objet dans un fichier
Pointer (Nom_logique, p)	Se pointer à une position p du fichier
Fin_fichier (Nom_logique)	Retourne Vrai si le pointeur est à la fin du fichier sinon elle retourne Faux
Taille_fichier (Nom_logique)	Retourne la taille du fichier
Fermer (Nom_logique)	Fermeture d'un fichier

b. Les fichiers textes

Algorithmique	Rôle
Ouvrir ("Chemin\Nom_physique", Nom_logique, "Mode")	Ouverture d'un fichier <ul style="list-style-type: none">• Mode d'ouverture :<ul style="list-style-type: none">○ "r" : Lecture○ "w" : Ecriture (création)○ "a" : Ecriture à la fin du fichier
Lire (Nom_logique, ch)	Lecture de la totalité d'un fichier
Lire_ligne (Nom_logique, ch)	Lecture d'une ligne depuis un fichier texte
Ecrire (Nom_logique, ch)	Ecriture dans un fichier texte
Fin_fichier (Nom_logique)	Retourne Vrai si le pointeur est à la fin du fichier sinon elle retourne Faux
Fermer (Nom_logique)	Fermeture d'un fichier

Recommandations :

- L'écriture de l'algorithme doit obligatoirement respecter l'**indentation**.
- La **nomenclature des objets** doit être significative.
- On utilise l'opérateur **+** pour **concaténer** deux chaînes.
- L'**indice du premier élément** d'une chaîne ou d'un tableau est **0**.
- La **valeur du pas** de la structure itérative complète peut être **positive ou négative**. Par défaut, elle est égale à 1.
- Les éléments d'un tableau doivent être de même type.
- Une fonction retourne **un seul résultat** de type simple (entier, réel, booléen, caractère, chaîne).
- Pour accéder à un élément d'un tableau à deux dimensions, on procède de la manière suivante : **Nom_objet [ligne, colonne]**.
- L'accès à un champ d'un enregistrement se fait de la manière suivante : **Nom_objet.champ**
- La **position initiale du pointeur dans un fichier de données** est **0**.