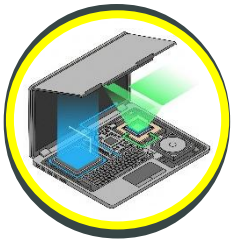
COURS D'

INFO



1ère Année



CONFORME AU PROGRAMME NATIONAL

EDITION 2025

TABLE DES MATIÈRES

PRÉLIMINAIRES **1**

1. Finalités et Buts du Cours 1
   - 1.1. Autonomie Numérique et Pensée Structurée 1
   - 1.2. Compétence Opérationnelle et Transversalité 1
   - 1.3. Fondement de la Citoyenneté Numérique 1

2. Objectif Terminal d'Intégration (OTI) 1
   - 2.1. Définition et Portée 1
   - 2.2. Situation d'Intégration Spécifique au Cours 2

3. Compétences de Base Attendues 2
   - 3.1. Maîtrise Conceptuelle et Lexicale 2
   - 3.2. Compétence Opérationnelle Fondamentale 2
   - 3.3. Sensibilisation aux Enjeux Numériques 2

PARTIE I : DE LA PENSÉE HUMAINE À LA LOGIQUE DE LA MACHINE **4**

Le cerveau, notre premier ordinateur 4

Donner des ordres clairs à une machine 4

Le langage secret des ordinateurs 4

Rendre les tâches automatiques et rapides 4

CHAPITRE 1 : FONDEMENTS DE L'ALGORITHMIQUE **5**

1.1. Programme et Langage de Programmation 5
   - a. L'ordre et la machine 5
   - A. L'Algorithme : La Recette de la Pensée 5
   - B. Le Programme : La Traduction pour la Machine 5

FIGURE 1 : RÉF. IMG (6682-5722-6441) **7**

Du Cerveau Humain à l'Écran : Le Processus de Création 7
   - C. Le Langage de Programmation : Le Dictionnaire Homme-Machine 7

1.2. Algorithme : La Recette de la Résolution de Problèmes 9
   - L'art de la recette, du Kwilu à l'ordinateur 9
   - A. Définition : L'algorithme comme plan d'action 10
   - B. Les caractéristiques d'une recette algorithmique 10

FIGURE 2 : RÉF. IMG (9236-1740-9271) **12**

Décomposition d'une tâche simple : la préparation du thé sucré 12

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

- L'anecdote de l'itinéraire imprécis.....12

1.3. Les Propriétés d'un Algorithme (Fini, Non-ambigu, Efficace).....14

- a. L'ordre secret d'une recette de cuisine.....14
- A. La Propriété "Fini" : Toute bonne chose a une fin.....14

FIGURE 3 : RÉF. IMG (4953-8980-1787).....16

Schéma d'un algorithme simple qui se termine après quelques étapes.....16

- B. La Propriété "Non-ambigu" : La machine n'interprète pas.....16
- C. La Propriété "Efficace" : Le chemin le plus direct.....16

1.4. Analyse d'un Problème : Données, Traitement et Résultats.....19

- a. La recette de cuisine, un algorithme sans le savoir.....19
- A. Les Données : La matière première.....19
- B. Le Traitement : Le moteur de la transformation.....20
- C. Les Résultats : Le produit fini.....20

FIGURE 4 : RÉF. IMG (1665-4803-3358).....22

Une scène de calcul et d'échange au marché de Matadi.....22

- b. Anecdote : La logique du chauffeur de bus à Goma.....22

1.5. Le Pseudocode : Structurer l'Idée en Langage Humain.....24

- a. Une recette pour la machine.....24
- A. Qu'est-ce que le Pseudocode ?.....24
- B. Les Mots-Clés de Base.....25

FIGURE 5 : RÉF. IMG (8838-3680-2625).....27

Du Dessin à l'Écriture : Représenter une Décision Simple.....27

FIGURE 6 : RÉF. IMG (0152-5066-6819).....29

Répéter une Tâche : la Boucle pour Compter des Mangues.....29

- a. Anecdote de l'agronome du Kwilu.....29

1.6. L'Ordinogramme : Visualiser le Flux d'Instructions.....31

- a. L'œil du chef de chantier.....31
- A. Le Langage des Formes : Symboles de Base.....31

FIGURE 7 : RÉF. IMG (8769-6889-5661).....34

Les symboles de base de l'ordinogramme et leur signification précise.....34

- B. Assembler les Idées : Du Problème au Schéma.....34
- Scénarios et Exercices.....35

1.7. Les Symboles Normalisés de l'Ordinogramme en Détail.....37

- a. Un langage universel pour les instructions.....37

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

- A. Le Début et la Fin : Le Symbole Ovale (Terminator).....37
- B. L'Action et le Traitement : Le Rectangle.....38
- C. La Question et la Décision : Le Losange.....38
- D. L'Entrée et la Sortie : Le Parallélogramme.....39

1.8. Exercices de Création d'Ordinogrammes Simples.....41

- a. L'erreur du dessinateur.....41
- A. La Logique Séquentielle : Le Chemin Droit.....41

FIGURE 8 : RÉF. IMG (3851-6284-4224).....43

Ordinogramme séquentiel simple pour préparer une tasse de thé.....43

- B. La Logique Conditionnelle : Le Carrefour de la Décision.....43

FIGURE 9 : RÉF. IMG (2638-8559-9268).....45

Ordinogramme conditionnel pour vérifier l'âge d'entrée dans un lieu.....45

- L'anecdote du marchand de beignets de Matadi.....45

PARTIE II : LES BRIQUES ÉLÉMENTAIRES D'UN PROGRAMME INFORMATIQUE.....48

a. Les instructions, des ordres pour une machine obéissante.....48

b. La variable, la mémoire à court terme du programme.....48

c. La condition, l'art de prendre une décision.....48

d. La boucle, le super-pouvoir de la répétition.....48

CHAPITRE 2 : MANIPULATION DES DONNÉES.....49

2.1. Variables et Constantes.....49

- a. La boîte de Maman.....49
- A. La Variable : la boîte à mémoire qui change.....49
- B. La Constante : la pierre gravée qui ne bouge pas.....49
- C. Pourquoi utiliser les deux ?.....50

2.2. Déclaration et Initialisation d'une Variable.....52

- a. Le principe des boîtes mémoires.....52
- A. La Déclaration : Réserver un Espace.....52

FIGURE 10 : RÉF. IMG (4069-5056-6637).....54

Schéma d'une variable déclarée mais non initialisée comme une boîte vide.....54

- B. L'Initialisation : Donner une Valeur Initiale.....54

FIGURE 11 : RÉF. IMG (4123-4535-8849).....57

Schéma d'une variable initialisée avec une valeur dans sa boîte.....57

- C. L'Affectation : Changer le Contenu.....57

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

2.3. Règles de Nommage des Variables et Constantes.....59
- a. Le pouvoir de nommer.....59
- A. Les Règles d'Or (Obligatoires).....59

FIGURE 12 : RÉF. IMG (9587-3979-8954).....62

Comparaison visuelle de noms de variables valides et invalides.....62
- B. Les Bonnes Habitudes (Conseillées).....62

FIGURE 13 : RÉF. IMG (5868-0561-6474).....64

Illustration des styles de nommage snake_case et camelCase.....64
- Une histoire de noms et d'étiquettes.....64

2.4. Les Types de Données de Base (Entier, Réel, Chaîne de caractères).....66
- a. L'ordinateur, un grand trieur.....66
- A. Les Nombres : Entiers et Réels.....67
- B. Le Texte : La Chaîne de Caractères.....67

FIGURE 14 : RÉF. IMG (1042-4271-0282).....69

Visualisation des trois types de données dans des boîtes étiquetées.....69
- L'inventaire de Papa Wendo.....69

2.5. Le Type de Donnée Booléen (Vrai/Faux).....71
- a. Le pari le plus simple du monde.....71
- A. Définition : Le Binaire au Cœur de la Décision.....71

FIGURE 15 : RÉF. IMG (1002-0708-2556).....73

L'interrupteur, un exemple parfait de logique booléenne : ON ou OFF.....73
- B. Utilité : Poser des Questions à l'Ordinateur.....73

2.6. Expressions et Opérateurs Arithmétiques.....75
- a. Le calcul mental du marché de Matadi.....75
- A. Les opérateurs, le cœur du calcul.....75
- B. La priorité des opérations.....76

FIGURE 16 : RÉF. IMG (1629-6633-7343).....78

Les cinq opérateurs arithmétiques fondamentaux en informatique.....78
- C. Construire une expression complexe.....78

2.7. Les Opérateurs de Comparaison.....80
- a. Comparer pour décider.....80
- A. Les Six Arbitres de la Logique.....80
- B. Le Verdict : Vrai ou Faux (Booléen).....81

FIGURE 17 : RÉF. IMG (7490-0211-1805).....83

Les opérateurs de comparaison illustrés par une balance de justice....83
- Une histoire de mangues à Kisangani....83

2.8. La Priorité des Opérateurs (Règles PEMDAS)....85
- a. La règle qui met de l'ordre dans le chaos....85
- A. L'Acronyme PEMDAS : La Hiérarchie des Calculs....85

FIGURE 18 : RÉF. IMG (8892-8771-7914)....87

Pyramide de la priorité des opérations selon la règle PEMDAS....87
- B. Application pratique : Décomposer un calcul....87

FIGURE 19 : RÉF. IMG (6099-0150-7442)....90

Le danger d'ignorer l'ordre : deux résultats pour un même calcul....90
- Une anecdote de cuisine pour tout comprendre....90

PARTIE III : DIALOGUE AVEC LA MACHINE : ENTRÉES, SORTIES ET INSTRUCTIONS....93

a. Le dialogue homme-machine....93

b. Donner des informations : les entrées....93

c. Recevoir les résultats : les sorties....93

d. Le langage du dialogue : les instructions....93

CHAPITRE 3 : LE FLUX D'EXÉCUTION SÉQUENTIEL....94

3.1. L'Instruction d'Affectation....94
- a. Une mémoire pour les données....94
- A. Le concept de variable et d'affectation....94

FIGURE 20 : RÉF. IMG (1166-0709-8686)....95

L'affectation comme une boîte de rangement pour les données de l'ordinateur....95
- B. Le processus d'exécution....95

3.2. L'Instruction d'Affichage (Sortie)....97
- a. L'ordinateur qui parle....97
- A. Le principe de la sortie....97

FIGURE 21 : RÉF. IMG (9202-0692-7919)....100

Schéma du flux de sortie : de l'instruction à l'écran....100
- B. Syntaxe de base : donner l'ordre d'afficher....100

FIGURE 22 : RÉF. IMG (4413-0589-9945)....103

Exemple de code d'affichage et son résultat sur un écran....103
- L'anecdote du distributeur de billets....103

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

3.3. Combiner Texte et Variables dans une Instruction d'Affichage.....105
- a. Le dialogue enrichi avec la machine.....105
- A. La Concaténation : Le principe du "collage".....105

FIGURE 23 : RÉF. IMG (4149-6929-3296).....107

Schéma de la concaténation de texte et de variables pour l'affichage.....107

FIGURE 24 : RÉF. IMG (2539-9811-9942).....109

Exemple de code affichant un message personnalisé avec nom et âge.....109
- Une anecdote pour mieux comprendre.....109

3.4. L'Instruction de Lecture (Entrée).....111
- a. Le dialogue nécessaire.....111
- A. Définition : L'Instruction d'Entrée.....111
- B. Le Processus en Trois Temps.....111

FIGURE 25 : RÉF. IMG (8467-1682-2164).....113

Visualisation d'une instruction de lecture en action sur un écran.....113
- Anecdote : Le recenseur du village de Masi-Manimba.....113

3.5. Gérer les Entrées Multiples de l'Utilisateur.....115
- a. L'interrogatoire séquentiel.....115
- A. Le principe de la file d'attente.....115

FIGURE 26 : RÉF. IMG (3901-0865-4573).....117

Diagramme du flux d'entrée séquentiel pour l'utilisateur.....117
- B. L'ordre des questions : une importance capitale.....117

FIGURE 27 : RÉF. IMG (3167-9466-0458).....119

Exemple de pseudo-code pour deux entrées successives.....119
- Le vendeur de billets de bus du Kongo Central.....119

3.6. Le Branchement Inconditionnel et la Séquentialité.....121
- a. Le chemin tracé et la bifurcation forcée.....121
- A. Le Principe Séquentiel : La Ligne Droite.....121

FIGURE 28 : RÉF. IMG (3960-9106-8828).....124

Schéma d'un algorithme séquentiel simple de haut en bas.....124
- B. La Rupture : Le Branchement Inconditionnel.....124

FIGURE 29 : RÉF. IMG (2569-0116-6524).....127

Schéma d'un algorithme avec un branchement inconditionnel (GOTO).....127
- L'anecdote : Le raccourci obligatoire à Boma.....127

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

3.7. Le Concept de 'Trace' d'un Programme pour le Débogage.....129
- a. Accroche : Le détective du code.....129
- A. Définir la Trace d'Exécution.....129

FIGURE 30 : RÉF. IMG (7533-4619-4298).....131

Visualisation d'une Trace d'Exécution à Côté de son Code Source.....131
- B. La Trace comme Outil de Débogage.....131

FIGURE 31 : RÉF. IMG (5577-1709-9870).....133

Débogage d'un Algorithme avec une Table de Trace Pas-à-Pas.....133
- Scénarios et Exercices.....133

3.8. Erreurs Courantes de Logique dans un Flux Séquentiel.....135
- a. L'ordre des actions.....135
- A. L'Erreur de Séquence : L'Ordre est Roi.....136

FIGURE 32 : RÉF. IMG (9031-8115-6338).....138

Schéma d'une erreur de logique dans une recette de cuisine simple.....138
- B. L'Oubli d'une Étape Cruciale.....138

FIGURE 33 : RÉF. IMG (5518-1902-3479).....141

Organigramme d'un calcul de salaire avec une étape manquante évidente.....141
- b. Scénarios et Exercices.....141

PARTIE IV : LE CŒUR DE LA LOGIQUE : LES STRUCTURES DE CONTRÔLE CONDITIONNELLES 144

Le principe du choix.....144

La condition, une question binaire.....144

Les structures SI… ALORS… SINON.....144

L'imbrication des choix multiples.....144

CHAPITRE 4 : LA PRISE DE DÉCISION.....145

4.1. La Structure Alternative Simple « Si… Alors… ».....145
- a. Le réflexe conditionnel.....145
- A. Définition et Syntaxe.....145

FIGURE 34 : RÉF. IMG (1653-1601-3846).....146

Schéma logique de la structure 'Si… Alors…' simple et son flux.....146

FIGURE 35 : RÉF. IMG (8050-6508-1465).....148

Algorithme concret pour la gestion d'un stock de pain à Kananga.....148

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

- b. Scénarios et Exercices.....148

4.2. La Structure Alternative Complète « Si… Alors… Sinon… ».....150

- a. Le choix à deux issues.....150
- A. Le principe du "plan B".....150
- B. L'embranchement sur la route logique.....150

FIGURE 36 : RÉF. IMG (7318-1475-7253).....152

Diagramme de la structure alternative complète Si… Alors… Sinon.....152

- L'anecdote du vendeur de beignets de Kananga.....152

4.3. Les Opérateurs Logiques « ET » et « OU ».....154

- a. Une logique du quotidien.....154
- A. L'Opérateur « ET » (AND) : La Condition Stricte.....154

FIGURE 37 : RÉF. IMG (9368-4377-2109).....156

Schéma simple de deux interrupteurs en série contrôlant une ampoule.....156

- B. L'Opérateur « OU » (OR) : La Condition Flexible.....156

FIGURE 38 : RÉF. IMG (9600-3430-7023).....158

Schéma de deux interrupteurs en parallèle contrôlant une ampoule.....158

- L'anecdote du chef de village de Kahemba.....158

4.4. L'Opérateur Logique de Négation « NON ».....160

- a. L'art du contraire.....160
- A. Le Principe de l'Inversion.....160
- B. La Table de Vérité du NON.....161

FIGURE 39 : RÉF. IMG (3615-2352-8301).....163

Le NON, l'interrupteur logique qui inverse le courant de la vérité.....163

- Scénarios et Exercices.....163
- Une histoire de rendez-vous à Goma.....163

4.5. Imbrication des Structures Conditionnelles (Si dans un Si).....165

- a. Les poupées russes de la logique.....165
- A. Définition et Principe.....165
- B. Représentation Visuelle : L'Organigramme.....165

FIGURE 40 : RÉF. IMG (7952-1765-4174).....167

Organigramme d'une structure conditionnelle SI imbriquée dans un autre SI.....167

- Scénarios et Exercices.....167
- Une histoire de transport de marchandises.....167

4.6. La Structure de Choix Multiple « Selon… Cas… ».....169

- a. L'efficacité cachée du quotidien.....169
- A. Principe du "Guichet Unique".....169

FIGURE 41 : RÉF. IMG (5813-3605-2671).....171

Organigramme de la structure de choix multiple, montrant un flux unique se di.....171

- B. Syntaxe Algorithmique.....171

FIGURE 42 : RÉF. IMG (5190-7598-7526).....173

Interface d'un distributeur de boissons : un exemple concret de choix multiple.....173

- L'anecdote de Madame la Secrétaire.....173

4.7. Introduction aux Fonctions Prédéfinies (INT, RND).....175

- a. Les outils secrets de l'ordinateur.....175
- A. La Fonction INT : Isoler la Partie Entière.....176

FIGURE 43 : RÉF. IMG (7431-6153-3511).....178

Visualisation de la fonction INT coupant la partie décimale d'un nombre.....178

- B. La Fonction RND : Créer de l'Aléatoire.....178

FIGURE 44 : RÉF. IMG (1897-3504-9745).....180

Générer un nombre aléatoire pour un jeu de dés virtuels.....180

- b. Scénarios et Exercices.....180

4.8. Exemples de Fonctions Prédéfinies Mathématiques et Textuelles.....182

- a. Les outils prêts à l'emploi de l'algorithme.....182
- A. Les fonctions mathématiques : les calculatrices de l'algorithme.....182

FIGURE 45 : RÉF. IMG (0849-9703-4220).....184

Un algorithme utilisant des fonctions mathématiques pour calculer un résultat.....184

- B. Les fonctions textuelles : les sculpteurs de mots.....184

FIGURE 46 : RÉF. IMG (2378-5504-7879).....186

La transformation d'une chaîne de caractères par des fonctions textuelles.....186

- b. Scénarios et Exercices.....186

ANNEXES.....188

1. Guide Pédagogique pour l'Enseignant.....188

- A. Philosophie et Objectifs Pédagogiques.....188
- B. Approche Méthodologique.....189
- C. Stratégies d'Évaluation.....189

2. Banque de Problèmes et de Projets.....190

- A. Projets basés sur les logiciels de bureautique.....190

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

- B. Problèmes d'analyse algorithmique et de logique (sans code).....193

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

PRÉLIMINAIRES

1. Finalités et Buts du Cours

1.1. Autonomie Numérique et Pensée Structurée

L'objectif premier est de doter l'élève d'une autonomie intellectuelle face aux outils numériques. Cette démarche dépasse la simple maîtrise fonctionnelle d'un logiciel. Elle vise à inculquer les principes de la pensée algorithmique : décomposer un problème complexe en sous-tâches simples, identifier des séquences logiques et anticiper des résultats. La finalité est de structurer la pensée de l'élève pour qu'il devienne acteur de la technologie, et non simple consommateur passif.

1.2. Compétence Opérationnelle et Transversalité

Au-delà de la théorie, le cours établit un socle de compétences techniques directement mobilisables. La maîtrise des outils bureautiques (traitement de texte, tableur) n'est pas une fin en soi. Elle constitue un levier de performance pour la réussite dans les autres disciplines (Français, Histoire, Géographie) et prépare l'élève aux réalités du monde professionnel futur, de l'Équateur au Kwilu. Le but est la production de travaux scolaires structurés et la résolution de problèmes concrets.

1.3. Fondement de la Citoyenneté Numérique

Enfin, ce programme initie à une conscience critique de l'environnement numérique. Les notions de propriété intellectuelle, de sécurité des données personnelles et de méthodologie de vérification de l'information en ligne sont introduites. L'objectif pédagogique est de former non seulement un utilisateur compétent, mais aussi un citoyen éclairé, conscient des enjeux, des droits et des devoirs liés à l'usage des technologies de l'information et de la communication dans la société congolaise et mondiale.

---

2. Objectif Terminal d'Intégration (OTI)

2.1. Définition et Portée

L'Objectif Terminal d'Intégration (OTI) formalise la compétence maîtresse que l'élève doit manifester au terme de l'année scolaire. Cette compétence exige la

mobilisation coordonnée de plusieurs savoirs, savoir-faire et savoir-être acquis durant les différentes leçons. L'élève résout une situation-problème complexe, authentique et significative, issue d'un contexte plausible. La validation de l'OTI atteste donc d'une maîtrise fonctionnelle des apprentissages et de leur transfert dans une situation nouvelle.

2.2. Situation d'Intégration Spécifique au Cours

Concrètement, pour ce cours d'informatique de première année, l'OTI consiste à **produire un document numérique simple et structuré en réponse à une tâche donnée**. Par exemple, face à une consigne précise (réaliser une fiche de présentation sur le parc national de la Salonga), l'élève doit de manière autonome : rechercher des informations, organiser un texte avec des titres, appliquer une mise en forme basique et gérer le fichier (nommer, enregistrer). L'évaluation de l'OTI se fonde sur la qualité du produit final et la pertinence de la démarche, non sur la simple exécution de tâches techniques disjointes.

---

3. Compétences de Base Attendues

3.1. Maîtrise Conceptuelle et Lexicale

Au terme de cette année, l'élève distingue avec précision les concepts fondamentaux de l'informatique. Il définit les notions de matériel (hardware) et de logiciel (software). Il identifie les principaux composants d'un ordinateur (processeur, mémoire vive, disque dur) et explique leurs rôles respectifs. Ce socle lexical lui permet de décrire un système informatique de manière structurée et cohérente, en utilisant un vocabulaire technique approprié.

3.2. Compétence Opérationnelle Fondamentale

La compétence pratique se manifeste par une manipulation assurée de l'environnement de travail numérique. L'élève gère l'arborescence des fichiers et des dossiers : création, copie, déplacement et suppression. Il démontre une autonomie fonctionnelle dans l'utilisation d'un logiciel de traitement de texte pour produire, mettre en forme et sauvegarder un document simple. L'objectif est l'efficacité opérationnelle sur des tâches bureautiques courantes.

3.3. Sensibilisation aux Enjeux Numériques

Le parcours de l'élève inclut une sensibilisation aux enjeux de la sécurité et de l'éthique. Il est capable d'identifier les menaces informatiques élémentaires

(virus, hameçonnage) et d'appliquer des règles de prudence de base. La notion de propriété intellectuelle et l'importance d'un comportement citoyen en ligne sont également introduites. Cette compétence prépare l'élève à une utilisation responsable et avertie des technologies de l'information.

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

Partie I : DE LA PENSÉE HUMAINE À LA LOGIQUE DE LA MACHINE

Le cerveau, notre premier ordinateur

Chaque jour, notre cerveau résout des problèmes complexes sans effort. Il nous permet de reconnaître un visage dans la foule à Matadi, de comprendre une histoire en kikongo ou de décider du meilleur moment pour planter du maïs au Kasaï. Cette intelligence naturelle combine la mémoire, le raisonnement et l'intuition. C'est le point de départ de toute notre exploration.

Donner des ordres clairs à une machine

Une machine, elle, fonctionne différemment. Elle ne possède pas notre bon sens. Pour lui faire accomplir une tâche, chaque étape doit être décrite avec une précision absolue, comme une recette de cuisine pour quelqu'un qui n'a jamais cuisiné. Oublier une seule instruction, même la plus simple, mène à l'échec. La machine exécute les ordres, elle ne les devine pas.

Le langage secret des ordinateurs

Notre logique humaine, formulée en français ou en swahili, est étrangère à la machine. L'ordinateur ne comprend qu'un seul langage : le binaire. Ce système est composé uniquement de deux chiffres, 0 et 1. Tout ce que fait un ordinateur, du simple calcul à l'affichage d'une photo de Mbandaka, est le résultat d'une immense combinaison de ces deux symboles.

Rendre les tâches automatiques et rapides

L'objectif final est de déléguer des tâches à la machine. En traduisant une logique en programme, nous créons un automatisme. L'ordinateur peut alors répéter une action des millions de fois, sans fatigue et sans erreur. Il peut gérer le stock d'une boutique à Kananga, calculer la trajectoire d'un avion ou simplement nous permettre de communiquer instantanément avec un proche au Kongo Central.

Chapitre 1 : FONDEMENTS DE L'ALGORITHMIQUE

1.1. Programme et Langage de Programmation

a. L'ordre et la machine

Un des faits les plus surprenants de l'informatique est la capacité d'une simple suite de mots à contrôler une machine complexe. Un smartphone, un ordinateur ou même un robot médical obéissent à des instructions écrites par un humain. À Matadi, un ordinateur gère les conteneurs du port ; à Mbuji-Mayi, il peut aider à trier les diamants.

Cette obéissance n'est pas magique. Elle provient d'un processus rigoureux où la pensée humaine est traduite en un langage que la machine peut exécuter. Comprendre ce processus est la première étape pour maîtriser l'informatique. Il s'agit de transformer une idée en action, en passant par une traduction stricte.

A. L'Algorithme : La Recette de la Pensée

Un algorithme est une suite finie et ordonnée d'instructions précises permettant de résoudre un problème ou d'accomplir une tâche. C'est un plan d'action. Avant de construire une maison, l'architecte dessine un plan. Avant d'écrire un programme, l'informaticien rédige un algorithme.

En cuisine, la recette pour préparer un bon plat de pondu est un algorithme. Elle liste les ingrédients (les données) et les étapes dans l'ordre (le traitement). Si les étapes sont claires et suivies correctement, le résultat est prévisible. L'algorithme est conçu pour un humain, il est exprimé en langage naturel.

B. Le Programme : La Traduction pour la Machine

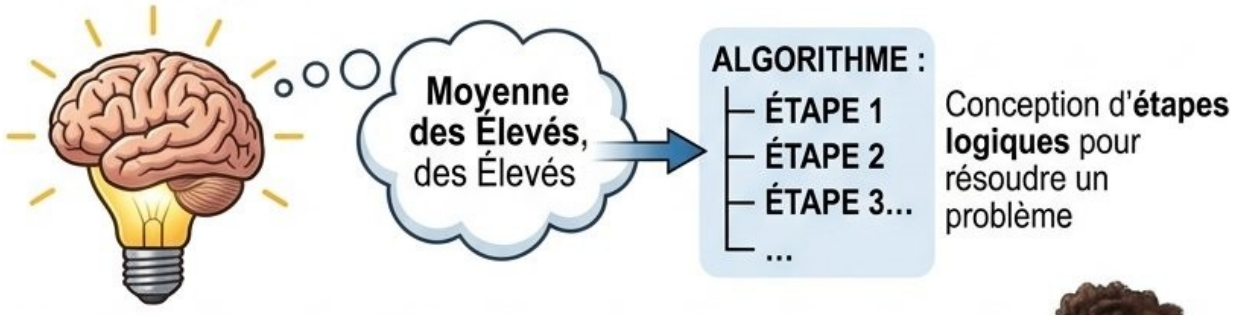
Un programme informatique est la concrétisation d'un algorithme dans un langage spécifique que l'ordinateur comprend. Il transforme la "recette" de l'algorithme en une série d'ordres que le processeur peut exécuter. Le programme est le plan de l'architecte traduit en instructions pour les maçons.

Fondamentalement, le programme est le texte écrit par le développeur. L'ordinateur lit ce texte et exécute les actions correspondantes. Sans programme, un ordinateur est une boîte inerte. C'est le programme qui lui donne une fonction : calculatrice, jeu vidéo, traitement de texte.

Figure 1 : Réf. IMG (6682-5722-6441)

Du Cerveau Humain à l'Écran : Le Processus de Création

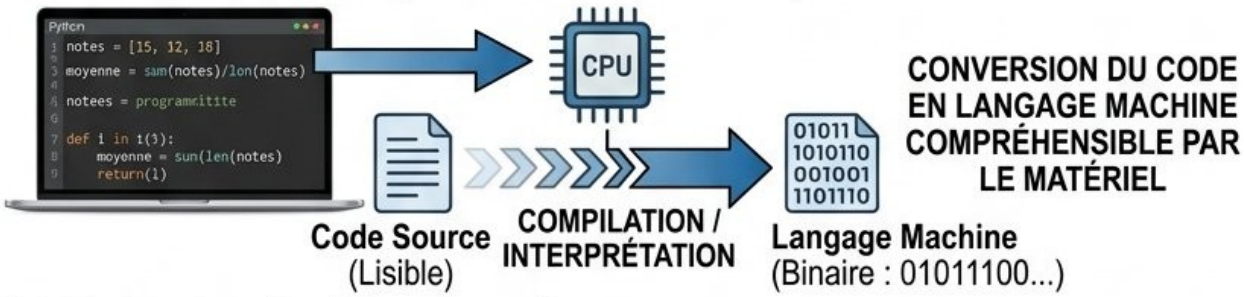
a. L'Idée (L'Algorithme)



b. L'Écriture (Le Langage)



c. La Traduction (Compilation/Interprétation)



d. L'Exécution (Le Programme)

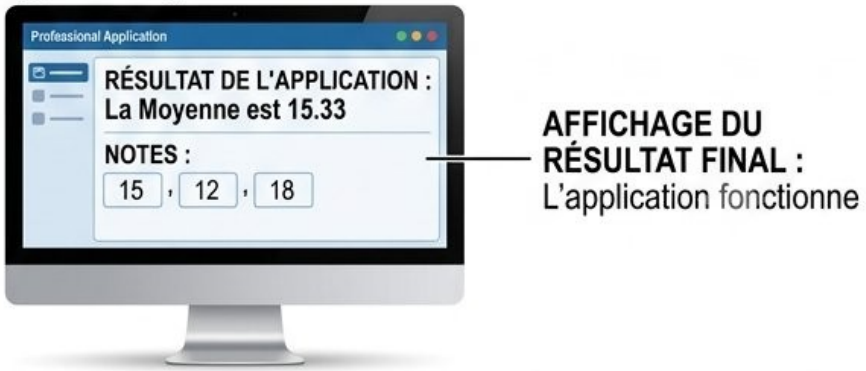


Figure 1 : Réf. IMG (6682-5722-6441)

Du Cerveau Humain à l'Écran : Le Processus de Création

a. L'Idée (L'Algorithme)

L'image montre un cerveau humain ou une ampoule. Cela symbolise la phase de réflexion pure. Ici, l'informaticien décompose un problème complexe, comme "calculer la moyenne des élèves", en étapes simples et logiques. C'est la conception de l'algorithme.

b. L'Écriture (Le Langage)

Ensuite, on voit une main qui écrit du code sur un clavier. C'est la phase de programmation. L'informaticien traduit chaque étape de l'algorithme en utilisant la grammaire et le vocabulaire d'un langage de programmation spécifique, comme Python ou Java.

c. La Traduction (Compilation/Interprétation)

Une flèche pointe de l'écran de code vers un symbole de processeur (CPU). Cette étape, souvent invisible pour l'utilisateur, est la traduction du code source en langage machine. C'est le moment où le langage "humain" du programmeur devient du binaire compréhensible par le matériel.

d. L'Exécution (Le Programme)

Enfin, l'image montre le résultat final sur un écran : une application qui affiche la moyenne calculée. Le programme est en action, il a pris les données d'entrée (les notes) et a produit le résultat attendu, suivant fidèlement les instructions de l'algorithme.

C. Le Langage de Programmation : Le Dictionnaire Homme-Machine

Un langage de programmation sert de pont entre la logique humaine et le fonctionnement binaire de l'ordinateur. Chaque langage possède son propre vocabulaire (les mots-clés) et sa propre grammaire (la syntaxe). Il en existe des centaines, chacun étant plus ou moins adapté à certaines tâches.

Comme le français, le lingala, le swahili ou le tshiluba permettent aux gens de communiquer, les langages comme Python, Java ou C++ permettent à un développeur de donner des ordres précis à l'ordinateur. Le choix du langage

dépend du projet : créer un site web, une application mobile ou un logiciel scientifique.

Anecdote : Le stock de Papa Kanku

Papa Kanku tient une boutique de pagnes très populaire à Kananga, au Kasaï. Chaque semaine, il passe des heures à recompter son stock à la main, avec un bic et un cahier. Il se trompe souvent, ce qui cause des problèmes avec ses clients. Sa nièce, étudiante en informatique à l'Université de Kananga, lui explique qu'un simple programme pourrait tout changer.

Elle lui décrit l'algorithme : "Pour chaque nouveau pagne qui arrive, on ajoute 1 au total. Pour chaque pagne vendu, on retire 1." L'idée est simple. Elle lui montre ensuite comment cela se traduirait dans un langage de programmation. Papa Kanku réalise alors qu'un "programme" n'est que la version machine d'une logique qu'il utilise déjà tous les jours. L'informatique démystifie le problème.

Exercice 1 : La liste de courses de Maman Hélène

Énoncé : Maman Hélène est au marché de Gambela à Kinshasa. Elle achète une botte de légumes à 1500 FC, un kilo de chèvre à 8000 FC et un litre d'huile à 4500 FC. Quel est l'algorithme pour calculer sa dépense totale ?

Résolution pas-à-pas :

1. **Étape 1 : Identifier les données (les entrées)** Le problème fournit trois données numériques : le prix des légumes (1500), le prix de la chèvre (8000) et le prix de l'huile (4500).
2. **Étape 2 : Définir l'algorithme** Un algorithme possible est :
   - Commencer avec un total de 0.
   - Ajouter le prix des légumes au total.
   - Ajouter le prix de la chèvre au total.
   - Ajouter le prix de l'huile au total.
   - Le résultat final est le total.
3. **Étape 3 : Appliquer l'algorithme** Total = 0 + 1500 = 1500 FC. Total = 1500 + 8000 = 9500 FC. Total = 9500 + 4500 = 14000 FC. La dépense totale est de 14000 FC. Le programme informatique automatiserait ce calcul.

Exercice 2 : Le bus de l'étudiant de Bukavu

Énoncé : Jean, étudiant à Bukavu, a 2000 FC dans sa poche. Le ticket de bus pour aller à l'université coûte 1200 FC. Quel est l'algorithme simple pour décider s'il peut prendre le bus ?

Résolution pas-à-pas :

1. **Étape 1 : Identifier les données et la question** Nous avons deux données : l'argent de Jean (2000 FC) et le prix du ticket (1200 FC). La question n'est pas un calcul, mais une décision (Oui/Non).
2. **Étape 2 : Définir l'algorithme de décision** L'algorithme est une comparaison :
   - SI l'argent en poche est supérieur ou égal au prix du ticket, ALORS la réponse est "Oui, il peut prendre le bus".
   - SINON, la réponse est "Non, il doit marcher ou trouver plus d'argent".
3. **Étape 3 : Appliquer l'algorithme** On compare 2000 FC et 1200 FC. L'affirmation "2000 est supérieur ou égal à 1200" est VRAIE. Donc, l'algorithme donne la réponse "Oui, il peut prendre le bus".

1.2. Algorithme : La Recette de la Résolution de Problèmes

L'art de la recette, du Kwilu à l'ordinateur

Pendant des siècles, la recette du plat de chenilles à la sauce tomate s'est transmise dans le Kwilu. Une pincée de sel en trop ou une minute de cuisson en moins, et le résultat change. La réussite dépend de la précision et de l'ordre des étapes.

L'informatique fonctionne sur ce même principe de rigueur. Une machine ne comprend pas l'à-peu-près. Pour lui demander de réaliser une tâche, il faut lui fournir une recette parfaite, claire et sans ambiguïté. Cette recette porte un nom : un algorithme.

A. Définition : L'algorithme comme plan d'action

Un algorithme est une suite finie et ordonnée d'instructions claires permettant de résoudre un problème donné. C'est la recette détaillée qui mène d'une situation de départ (le problème) à un résultat souhaité (la solution).

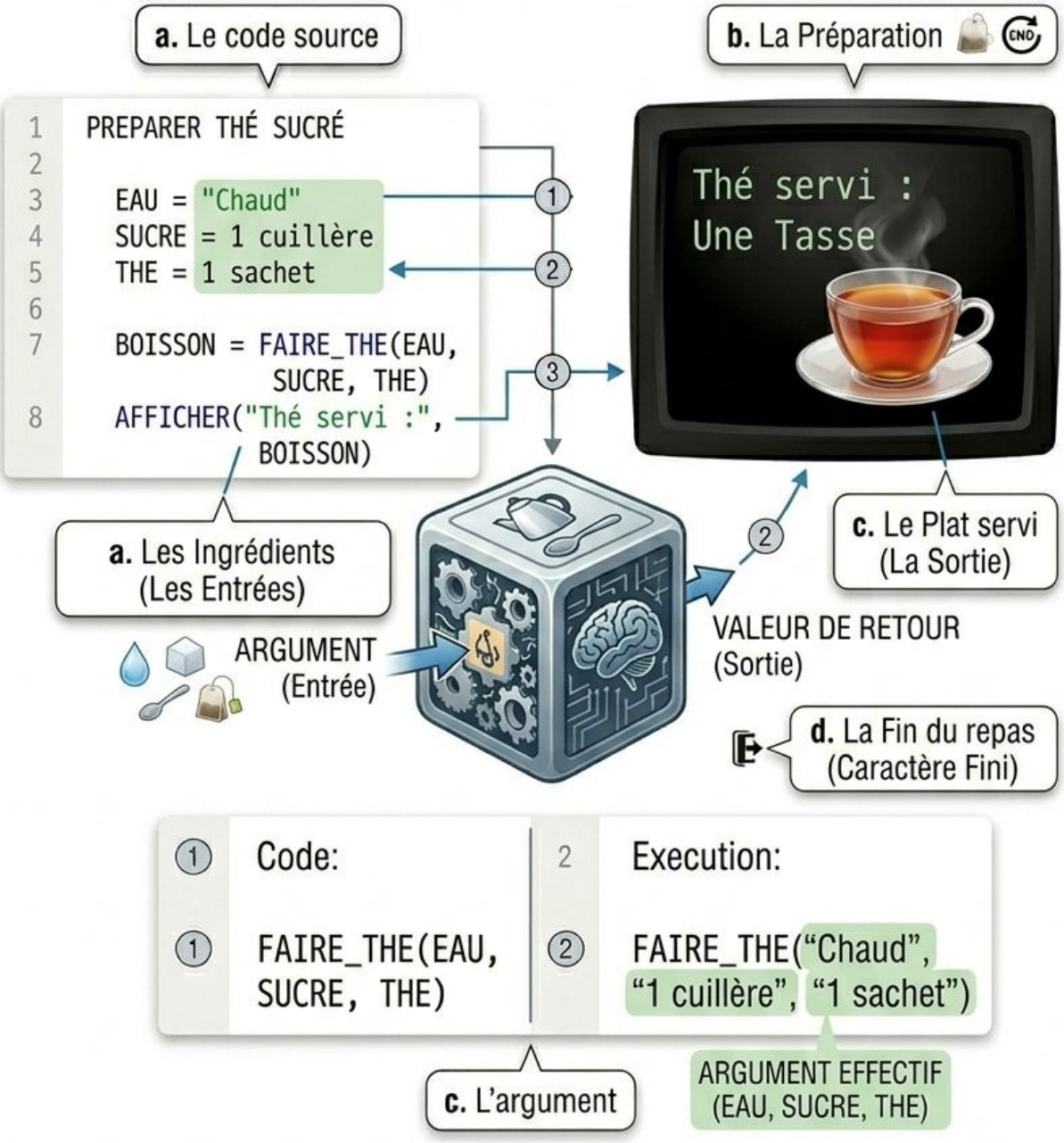
Avant de construire une maison à Lubumbashi, l'architecte dessine un plan détaillé. Ce plan est un algorithme. Le maçon, en suivant les étapes du plan, exécute l'algorithme. Le maçon n'a pas besoin de deviner ; il suit les instructions. L'ordinateur est un maçon très rapide, mais sans aucune imagination. Il a besoin d'un plan parfait.

B. Les caractéristiques d'une recette algorithmique

Toute bonne recette possède des qualités précises. Il en va de même pour un algorithme. Pour qu'il soit compréhensible par une machine (ou par une autre personne), il doit respecter trois points fondamentaux.

Figure 2 : Réf. IMG (9236-1740-9271)

Décomposition d'une tâche simple :
la préparation du thé sucré



1	Code:	2	Execution:
1	FAIRE_THE(EAU, SUCRE, THE)	2	FAIRE_THE("Chaud", "1 cuillère", "1 sachet")

c. L'argument

ARGUMENT EFFECTIF (EAU, SUCRE, THE)

Figure 2 : Réf. IMG (9236-1740-9271)

Décomposition d'une tâche simple : la préparation du thé sucré

a. Les Ingrédients (Les Entrées)

Un algorithme a besoin de données pour travailler. Pour faire du thé, il faut de l'eau, du sucre, et un sachet de thé. Ce sont les "entrées" de l'algorithme. Sans elles, la recette ne peut pas commencer.

b. La Préparation (Le Traitement)

Les instructions doivent être précises et sans ambiguïté. "Chauffer l'eau", "Verser sur le sachet", "Ajouter le sucre". Chaque étape est une action simple et réalisable. C'est le cœur de l'algorithme, la transformation des entrées.

c. Le Plat servi (La Sortie)

Un algorithme produit toujours un résultat. Le résultat de la recette du thé est une tasse de thé chaud et sucré. C'est la "sortie" de l'algorithme. Il apporte la solution au problème initial.

d. La Fin du repas (Caractère Fini)

Une recette a une fin. L'algorithme doit se terminer après un nombre défini d'étapes. Une recette qui demanderait de remuer le thé à l'infini ne serait pas une bonne recette.

L'anecdote de l'itinéraire imprécis

Papa Kabasele, fraîchement arrivé de Kananga pour visiter sa famille à Kinshasa, demande son chemin pour aller au Grand Marché. Son neveu, un peu distrait, lui répond : "Tonton, c'est facile ! Tu sors, tu marches un peu tout droit, et quand tu vois un grand panneau publicitaire, tu tournes sur la grande avenue. Tu ne peux pas le rater !".

Papa Kabasele sort. Il voit trois grands panneaux publicitaires différents et deux avenues qui lui semblent "grandes". Après une heure de marche infructueuse sous le soleil, il est complètement perdu. L'instruction de son

neveu était un mauvais algorithme : elle n'était ni précise ("marcher un peu", "grand panneau"), ni unique. Une machine aurait connu le même sort. Pour un ordinateur, "tourner à la troisième rue à droite et continuer sur 200 mètres" est une instruction parfaite.

Exercice 1 : La somme des points de Mboyo

Énoncé : Mboyo, élève à Kisangani, a obtenu les points suivants en interrogation : 7 en maths, 8 en français, 5 en histoire. Il veut connaître son total de points. Rédige l'algorithme simple pour le calculer.

Résolution pas-à-pas :

1. **Étape 1 : Identifier les entrées** Les données de départ sont les notes obtenues : 7, 8 et 5.
2. **Étape 2 : Définir le traitement** L'opération nécessaire pour trouver le total est une addition.
3. **Étape 3 : Exécuter les instructions** Prendre la première note (7). Ajouter la deuxième note au résultat (7 + 8 = 15). Ajouter la troisième note au résultat (15 + 5 = 20).
4. **Étape 4 : Annoncer la sortie** Le résultat final est 20. Le total de points de Mboyo est 20.

Exercice 2 : La caisse de Mama Elodie

Énoncé : Mama Elodie tient une petite boutique à Mbuji-Mayi. Un client achète un pain de 500 Francs Congolais (FC) et lui donne un billet de 1000 FC. Quel est l'algorithme pour calculer la monnaie à rendre ?

Résolution pas-à-pas :

1. **Étape 1 : Identifier les entrées** Les données sont le prix de l'article (500 FC) et le montant donné par le client (1000 FC).
2. **Étape 2 : Définir le traitement** L'opération pour calculer la monnaie est une soustraction : montant donné moins le prix.
3. **Étape 3 : Exécuter les instructions** Prendre le montant donné (1000). Soustraire le prix de l'article (1000 - 500 = 500).
4. **Étape 4 : Annoncer la sortie** Le résultat est 500. Mama Elodie doit rendre 500 FC au client.

1.3. Les Propriétés d'un Algorithme (Fini, Non-ambigu, Efficace)

a. L'ordre secret d'une recette de cuisine

Même si une recette de cuisine pour le pondu semble simple, elle doit respecter des règles précises pour réussir. Oublier une étape, utiliser des instructions vagues ("ajoute un peu de sel") ou tourner en rond sans jamais finir la cuisson peut transformer un plat délicieux en un échec total. Une bonne recette est, sans le savoir, un bon algorithme.

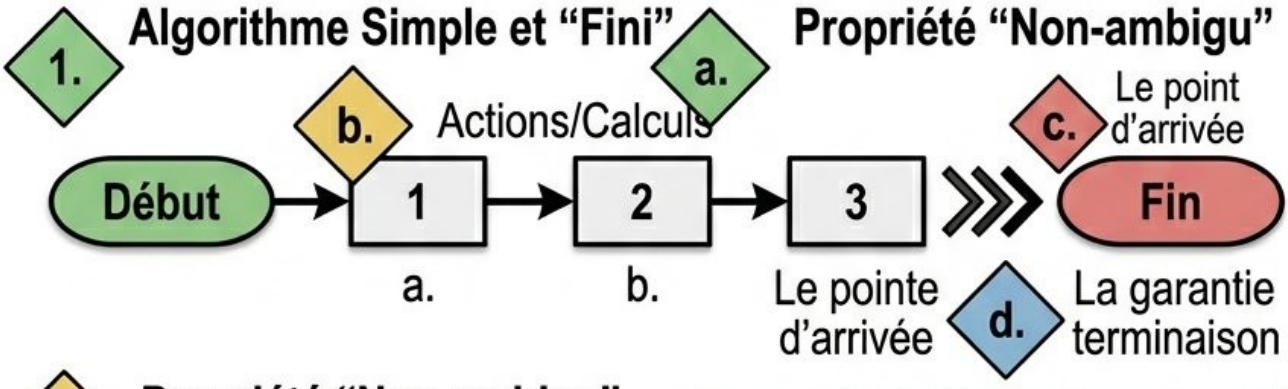
Un algorithme informatique fonctionne exactement de la même manière. Pour qu'une machine puisse l'exécuter sans erreur, il doit posséder des qualités fondamentales. Ces propriétés garantissent que l'ordinateur comprendra la tâche, l'exécutera correctement et s'arrêtera une fois le travail terminé. Trois qualités sont essentielles.

A. La Propriété "Fini" : Toute bonne chose a une fin

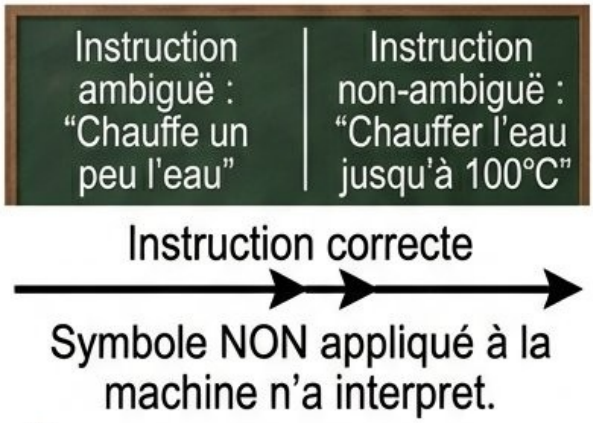
Un algorithme doit obligatoirement se terminer après un nombre fini d'étapes. Une machine ne peut pas exécuter une série d'instructions qui continue à l'infini. Cela provoquerait une "boucle infinie", un des bugs les plus courants en programmation. Chaque cheminement logique doit avoir un point de départ et, surtout, un point d'arrivée clair.

Penser à un voyage en train de Kananga vers Ilebo. Le voyage a un début (la gare de Kananga) et une fin (la gare d'Ilebo). Un algorithme qui ne finit jamais est comme un train qui roulerait sur des rails en cercle sans jamais atteindre une destination.

Figure 3 : Réf. IMG (4953-8980-1787)
Schéma d'un algorithme simple qui se termine après quelques étapes





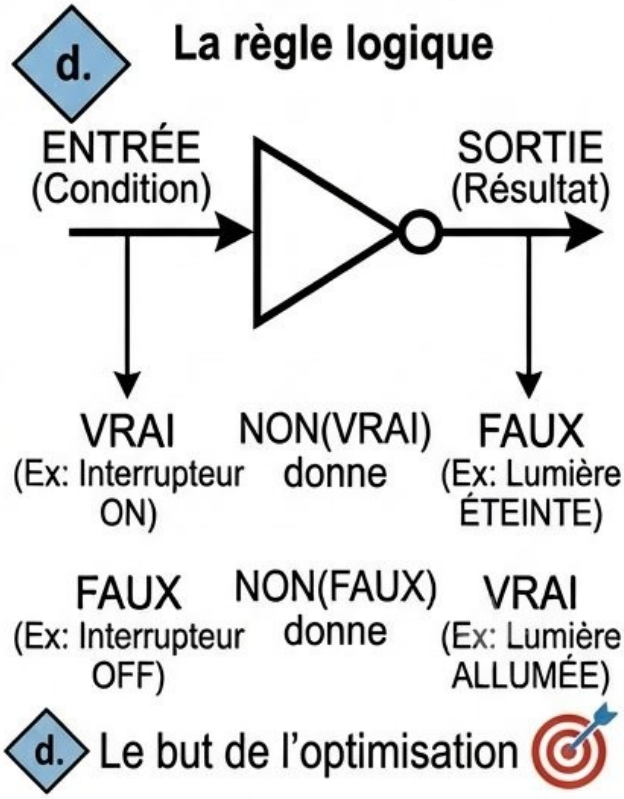






c. L'économie des ressources

d. Le but de l'optimisation



d. Le but de l'optimisation

Figure 3 : Réf. IMG (4953-8980-1787)

Schéma d'un algorithme simple qui se termine après quelques étapes.

a. Le point de départ

Le processus commence toujours par une instruction de départ claire, ici matérialisée par l'ovale "Début". C'est le signal initial que l'ordinateur attend pour commencer son travail. Sans ce point de départ, aucune action ne peut être initiée.

b. Une suite d'actions

Les rectangles représentent des actions ou des calculs précis. Chaque étape est exécutée l'une après l'autre. L'ordre est crucial. La machine suit le chemin indiqué par les flèches sans jamais dévier, garantissant une exécution prévisible et contrôlée.

c. Le point d'arrivée

L'ovale "Fin" est l'instruction la plus importante pour cette propriété. Il signifie que le travail est accompli. L'algorithme a rempli sa mission et libère les ressources de la machine pour d'autres tâches.

d. La garantie de terminaison

La présence de cet élément "Fin" garantit que le processus n'est pas infini. C'est la preuve visuelle que l'algorithme est "fini".

B. La Propriété "Non-ambigu" : La machine n'interprète pas

Chaque instruction d'un algorithme doit être claire, précise et ne laisser aucune place à l'interprétation. Un ordinateur est une machine logique, pas un être humain. Il exécute les ordres à la lettre. Une instruction comme "chauffe un peu l'eau" est ambiguë. Faut-il chauffer à 40°C ou à 90°C ?

L'instruction correcte serait : "Chauffer l'eau jusqu'à atteindre 100°C". Dans ce cas, il n'y a qu'une seule façon de comprendre et d'exécuter l'ordre. Chaque étape doit être définie sans la moindre ambiguïté.

C. La Propriété "Efficace" : Le chemin le plus direct

Un algorithme doit résoudre un problème en utilisant le minimum de ressources possibles (temps de calcul, mémoire de l'ordinateur). Il existe

souvent plusieurs manières de résoudre un même problème. Le bon algorithme est celui qui y parvient le plus rapidement et le plus économiquement.

[IMAGE: Carte montrant deux chemins pour relier deux villes congolaises.]

a. Le trajet inefficace

Un premier chemin sinueux et plein de détours relie deux points. Il représente un algorithme qui fonctionne, mais qui prend beaucoup trop de temps ou de mémoire. Il fait des calculs inutiles avant d'arriver au résultat.

b. Le trajet efficace

Un second chemin, beaucoup plus direct, relie les mêmes deux points. C'est l'image d'un algorithme efficace. Il atteint le même objectif, mais en utilisant une méthode plus intelligente, plus rapide et qui consomme moins de ressources.

c. L'économie des ressources

En informatique, le temps et la mémoire sont comme le carburant d'une voiture. Le chemin direct économise du carburant. Un algorithme efficace économise la puissance de l'ordinateur, le rendant plus rapide pour l'utilisateur.

d. Le but de l'optimisation

L'objectif de tout bon programmeur est de trouver le chemin le plus efficace, comme un ingénieur routier qui conçoit une route droite entre Bukavu et Goma pour réduire le temps de trajet.

b. Scénarios et Exercices

L'artisan de la Province Orientale qui fabrique une pirogue suit, sans le savoir, un algorithme ancestral. Il commence par choisir le bon arbre (début), le creuse selon des techniques précises (instructions non-ambiguës), puis s'arrête quand la pirogue flotte correctement (fin). S'il sculptait le bois sans fin, la pirogue coulerait. Si ses gestes étaient ambigus, la forme serait ratée. Son savoir-faire est un algorithme efficace, transmis de génération en génération. Changer une seule étape, c'est risquer de rendre la pirogue inutilisable. L'informatique s'inspire de cette même rigueur : chaque instruction compte. Un seul mot vague ou une étape manquante peut faire "couler" tout le programme. La machine, comme le bois, ne pardonne pas l'imprécision.

Exercice 1 : La recette de Mama Hélène

Énoncé : Mama Hélène a deux recettes pour préparer le fufu. Identifie celle qui représente un algorithme correct et explique pourquoi en te basant sur les trois propriétés vues en cours.

- **Recette A :** "Prendre de la farine de manioc. Ajouter de l'eau jusqu'à obtenir une bonne consistance. Tourner la pâte pendant un bon moment sur le feu. Servir quand c'est prêt."
- **Recette B :** "1. Mettre 1 litre d'eau dans une marmite. 2. Porter à ébullition. 3. Verser 500g de farine de manioc. 4. Tourner vigoureusement pendant 15 minutes. 5. Arrêter le feu. 6. Servir la pâte."

Résolution pas-à-pas :

1. **Étape 1 : Analyser la Recette A** Cette recette est incorrecte. Elle n'est pas un bon algorithme.
   - Elle est **ambiguë** : "bonne consistance", "bon moment", "quand c'est prêt" sont des termes vagues.
   - Elle n'a pas de critère de fin clair, ce qui la rend potentiellement **infinie**.
2. **Étape 2 : Analyser la Recette B** Cette recette est un algorithme correct.
   - Elle est **finie** : Le processus s'arrête à l'étape 6.
   - Elle est **non-ambiguë** : Chaque quantité (1 litre, 500g) et durée (15 minutes) est précise.
   - Elle est **efficace** : Les étapes sont logiques et directes pour obtenir le résultat.

Exercice 2 : Le chauffeur de taxi-moto de Boma

Énoncé : Un client appelle un chauffeur de taxi-moto à Boma et lui donne ces instructions : "Je suis devant une grande parcelle sur l'avenue principale, pas loin du marché. Viens me chercher." Le chauffeur risque de se perdre. Réécris ces instructions pour qu'elles deviennent non-ambiguës.

Résolution pas-à-pas :

1. **Étape 1 : Identifier les ambiguïtés** Les instructions originales sont vagues :
   - "grande parcelle" : Il peut y en avoir plusieurs.
   - "avenue principale" : Laquelle ? Boma en a plusieurs.

- "pas loin du marché" : C'est une distance relative et imprécise.

2. **Étape 2 : Rédiger des instructions non-ambiguës** Une version algorithmique correcte serait :
   - "1. Prends l'Avenue Lumumba en direction du marché central."
   - "2. Après le rond-point, compte trois parcelles sur ta droite."
   - "3. Arrête-toi devant la parcelle au portail vert, numéro 127."
   - "4. Je suis habillé avec une chemise jaune." Ces instructions sont précises et ne permettent qu'une seule interprétation.

1.4. Analyse d'un Problème : Données, Traitement et Résultats

a. La recette de cuisine, un algorithme sans le savoir

Lorsque une maman de Mbandaka prépare un délicieux plat de pondu, elle ne le fait pas au hasard. Elle rassemble d'abord ses ingrédients : les feuilles de manioc, l'huile de palme, le poisson fumé. Ensuite, elle suit des étapes précises : piler, cuire, assaisonner. Finalement, elle obtient un plat savoureux prêt à être servi.

Cette logique est exactement celle que l'informatique utilise pour résoudre un problème. Tout problème informatique se décompose en trois parties fondamentales. Il faut des informations de départ, une série d'actions à exécuter, et un résultat final à présenter.

A. Les Données : La matière première

Les données sont les informations brutes nécessaires pour commencer à résoudre un problème. Elles sont le point de départ de tout algorithme. Sans données, aucun traitement n'est possible. Ces données peuvent être des nombres, du texte, des images ou des sons. Pour calculer la moyenne d'un élève, les données sont ses différentes notes (15/20, 12/20, 09/20).

[IMAGE: Un agriculteur du Kasaï triant sa récolte de haricots]

a. L'état brut

L'image montre une grande quantité de haricots fraîchement récoltés. Cet amas représente les données brutes, non organisées. On y trouve des bons grains, des mauvais, et peut-être des petites pierres. L'ordinateur reçoit souvent des informations dans cet état initial.