cours d'
INFOMATIQUE

2è Année



CONFORME
AU PROGRAMME
NATIONAL

EDITION
2025

TABLE DES MATIÈRES

PRÉLIMINAIRES **1**

1. Finalités et Buts du Cours 1
   - 1.1. Développement d'une Compétence Instrumentale 1
   - 1.2. Structuration d'une Pensée Logique et Analytique 1
   - 1.3. Instauration d'une Conscience Citoyenne Numérique 1

2. Objectif Terminal d'Intégration (OTI) 1
   - 2.1. Description de la compétence 2
   - 2.2. Contexte de la situation d'intégration 2
   - 2.3. Productions attendues de l'élève 2

3. Compétences de Base Attendues 2
   - 3.1. Maîtrise Instrumentale et Opérationnelle 2
   - 3.2. Initiation au Raisonnement Logique et Algorithmique 3
   - 3.3. Culture Numérique et Citoyenneté 3

PARTIE 1. L'AUTOMATISATION DES TÂCHES RÉPÉTITIVES : LES BOUCLES **4**

La répétition, une punition humaine 4

L'ordinateur, un travailleur infatigable 4

Donner l'ordre de répéter : la boucle 4

L'enjeu : la puissance et le gain de temps 4

CHAPITRE 1. LES STRUCTURES ITÉRATIVES **5**

1.1. Le Principe de la Répétition Contrôlée 5
   - a. L'intelligence de la paresse 5
   - A. La boucle : le moteur de la répétition 5
   - B. La condition de sortie : le secret du contrôle 5

FIGURE 1 : RÉF. IMG (7221-8963-2594) **7**

Schéma simple d'une boucle contrôlée par une condition de sortie 7
   - C. Types de boucles courantes 7
   - SCÉNARIOS ET EXERCICES 8

1.2. La Boucle à Comptage « Pour… De… À… » 9
   - a. Une tâche prévisible 9
   - A. Définition et Syntaxe 9
   - B. Le Mécanisme du Compteur 10

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

FIGURE 2 : RÉF. IMG (2810-7545-2744) **12**

Schéma du fonctionnement d'une boucle à comptage avec son compteur 12

- b. L'appel des finalistes au TENASOSP 12

1.3. Anatomie d'une Boucle : Initialisation, Condition, Incrémentation 14

- a. Le secret d'une tâche bien répétée 14
- A. L'Initialisation : Le Point de Départ 14
- B. La Condition : Le Gardien de la Boucle 15

FIGURE 3 : RÉF. IMG (0347-2776-7627) **17**

Les trois piliers d'une boucle : Départ, Test et Progression 17

- C. L'Incrémentation : Le Moteur du Progrès 17

1.4. La Boucle Conditionnelle « Tant que… Faire… » 19

- a. Le principe de la vérification préalable 19
- A. Anatomie de la boucle « Tant que… » 20

FIGURE 4 : RÉF. IMG (8583-1576-5809) **22**

Schéma logique de la boucle "Tant que" avec sa condition d'entrée 22

- B. Le risque de la boucle infinie 22

FIGURE 5 : RÉF. IMG (4763-6378-0059) **25**

Comparaison d'une boucle correcte et d'une boucle infinie 25

- C. Scénarios et Exercices 25

1.5. La Boucle Conditionnelle « Répéter… Jusqu'à… » 27

- a. La logique du fait accompli 27
- A. Principe de la Boucle "Répéter… Jusqu'à…" 27

FIGURE 6 : RÉF. IMG (5789-6341-6794) **29**

Schéma de fonctionnement de la boucle Répéter Jusqu'à 29

- B. Syntaxe en Pseudocode 29

FIGURE 7 : RÉF. IMG (1857-8848-0064) **31**

Comparaison visuelle entre les boucles Tant que et Répéter 31

- Anecdote : La patience du pêcheur de Mbandaka 31

1.6. Choisir la Bonne Boucle : « Pour » vs « Tant que » 34

- a. Le dilemme du programmeur 34
- A. La Boucle Pour : Le Compteur Fiable 34
- B. La Boucle Tant que : Le Gardien de Condition 34

FIGURE 8 : RÉF. IMG (1232-8135-3040) **36**

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

La course des boucles : le compteur contre le gardien de condition....36
- C. Tableau récapitulatif....36

1.7. Les Boucles Imbriquées : Répétitions dans la Répétition....38
- a. Le cycle dans le cycle....38
- A. Anatomie d'une Boucle Imbriquée....38

FIGURE 9 : RÉF. IMG (3970-8397-0414)....41

Schéma visuel du fonctionnement d'une boucle imbriquée avec deux compteurs....41
- B. Applications concrètes....41

FIGURE 10 : RÉF. IMG (6539-9300-7630)....43

Génération d'une table de multiplication grâce à des boucles imbriquées....43
- SCÉNARIOS ET EXERCICES....43

1.8. Erreurs Courantes : La Boucle Infinie et l'Erreur d'Indice....45
- a. L'erreur du premier programmeur....45
- A. La Boucle Infinie : Le piège de l'éternité....45

FIGURE 11 : RÉF. IMG (8483-2187-1039)....47

Visualisation d'une boucle infinie comparée à une boucle normale....47
- B. L'Erreur d'Indice : Le pas de trop....47

PARTIE 2. LA GESTION DE COLLECTIONS DE DONNÉES : LES TABLEAUX....50

a. La mémoire d'un seul nom....50

b. L'idée du casier....50

c. Le tableau, une liste numérotée....50

d. L'organisation à grande échelle....50

CHAPITRE 2. LES VARIABLES INDICÉES (TABLEAUX)....51

2.1. De la Variable Simple au Tableau : Le Besoin d'Organiser....51
- a. Le casse-tête de l'enseignant....51
- A. La Limite de la Variable Simple....51
- B. L'Avènement du Tableau : une Collection Ordonnée....51

FIGURE 12 : RÉF. IMG (4595-0266-9800)....54

De la Variable Isolée au Tableau Structuré....54
- Le Comptoir de Maman Olangi....54

2.2. Déclaration d'un Tableau : Nom, Type et Dimension....56
- a. Organiser pour mieux régner....56
- A. Les trois piliers de la déclaration....56

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

FIGURE 13 : RÉF. IMG (4959-5720-8217)..... 59

Décomposition visuelle de la déclaration d'un tableau en pseudo-code..... 59

- B. Syntaxe de la déclaration en pseudo-code..... 59

FIGURE 14 : RÉF. IMG (3118-6972-5353)..... 62

Comparaison entre un tableau à une dimension et à deux dimensions..... 62

- C. La dimension : simple ou multiple ?..... 62

2.3. Accès aux Éléments : L'Importance de l'Indice..... 64

- a. Le principe de l'adresse unique..... 64
- A. La convention de l'indice zéro..... 64

FIGURE 15 : RÉF. IMG (6512-9965-8041)..... 67

Un tableau informatique simple avec ses indices et ses valeurs..... 67

- B. La syntaxe d'accès : le rôle des crochets..... 67

FIGURE 16 : RÉF. IMG (5796-4376-1745)..... 69

Schéma montrant la lecture et l'écriture dans une case de tableau..... 69

- Anecdote : Les paniers de Maman Alphonsine..... 69

2.4. Initialisation d'un Tableau : Remplissage à la Déclaration..... 71

- a. Le gain de temps, un principe universel..... 71
- A. Le principe de l'initialisation..... 71

FIGURE 17 : RÉF. IMG (6023-3666-6890)..... 73

Remplissage : méthode classique versus méthode d'initialisation à la déclarat..... 73

- B. La syntaxe en pseudo-code..... 73

FIGURE 18 : RÉF. IMG (8128-4589-3459)..... 76

Anatomie d'une déclaration de tableau initialisé en pseudo-code..... 76

- Scénarios et Exercices..... 76
- L'inventaire de Maman Hélène à Mbandaka..... 76

2.5. Parcours Complet d'un Tableau avec une Boucle « Pour »..... 78

- a. L'appel nominal comme algorithme..... 78
- A. Le Principe : Un Itérateur et Trois Commandes..... 78

FIGURE 19 : RÉF. IMG (1469-2572-6920)..... 80

Visualisation d'une boucle parcourant un tableau de notes d'élèves..... 80

- B. Syntaxe en Pseudocode..... 80
- C. Scénarios et Exercices..... 81

2.6. Opérations Courantes : Recherche d'un Élément dans un Tableau..... 83

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

- a. L'appel du matin....83
- A. La Recherche Séquentielle : Une Méthode Logique....83

FIGURE 20 : RÉF. IMG (6426-3642-8152)....84

Visualisation d'une recherche séquentielle dans un tableau numérique....84

- B. Les Deux Issues Possibles : "Trouvé" ou "Non Trouvé"....84

FIGURE 21 : RÉF. IMG (5428-8660-8445)....86

Les deux résultats possibles de la recherche de l'élément "Mangue"....86

- Anecdote : Le colis perdu à l'entrepôt de Matadi....86

2.7. Opérations Courantes : Calcul de la Somme et de la Moyenne des Éléments....88

- a. Introduction : Du marché de Mbandaka au processeur....88
- A. Calcul de la somme des éléments....88

FIGURE 22 : RÉF. IMG (8488-9104-4232)....90

Algorithme visuel du calcul de la somme d'un tableau....90

- B. Calcul de la moyenne des éléments....90

FIGURE 23 : RÉF. IMG (7629-4182-6573)....93

Logique de calcul de la moyenne à partir de la somme....93

- b. Scénarios et Exercices....93

2.8. Les Tableaux à Deux Dimensions (Matrices) : Introduction....95

- a. Le besoin d'une structure en grille....95
- A. Définition et Analogie Visuelle....95

FIGURE 24 : RÉF. IMG (9972-8268-8928)....97

Représentation d'une matrice comme une grille de places dans un bus....97

- B. Déclaration et Accès aux Éléments....97

FIGURE 25 : RÉF. IMG (0807-0298-3347)....100

Un calendrier mensuel : la matrice parfaite de la vie quotidienne....100

- b. Scénarios et Exercices....100

PARTIE 3. LA PROGRAMMATION STRUCTURÉE : FONCTIONS ET SOUS-PROGRAMMES. .102

a. La limite du code en "longue liane"....102

b. L'intelligence du "diviser pour régner"....102

c. Fabriquer ses propres outils....102

d. Fonctions et sous-programmes : des missions spéciales....102

CHAPITRE 3. LA MODULARISATION DU CODE....103

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

3.1. Le Principe du « Diviser pour Régner »....103
- a. Une stratégie militaire devenue un principe informatique....103
- A. Découper un grand problème....103

FIGURE 26 : RÉF. IMG (5036-0387-7874)....105

Schéma montrant un code monolithique devenant un code modulaire organisé....105
- B. Le programme principal comme chef d'orchestre....105

3.2. Définition et Appel d'une Fonction Simple....107
- a. Le principe du "raccourci" intelligent....107
- A. La Définition : Créer sa propre "Boîte à Outils"....108

FIGURE 27 : RÉF. IMG (1914-1208-0564)....110

Anatomie d'une fonction : une boîte à outils nommée et réutilisable....110
- B. L'Appel : Activer la "Boîte à Outils"....110

FIGURE 28 : RÉF. IMG (0705-8630-0205)....113

Le mécanisme d'appel : le programme principal délègue une tâche spécifique....113
- L'astuce de Papa Kabasele au "Wenze ya Kwilu"....113

3.3. Le Passage de Paramètres : Communiquer avec les Fonctions....115
- a. La commission de l'oncle....115
- A. Paramètres et Arguments : Le Contrat de la Fonction....115

FIGURE 29 : RÉF. IMG (6562-9736-2407)....117

Schéma d'une fonction recevant un paramètre et retournant une valeur....117
- B. Deux Méthodes de Transmission....117

FIGURE 30 : RÉF. IMG (8618-8975-7549)....120

Comparaison visuelle entre le passage par valeur et par référence....120
- C. Scénarios et Exercices....120

3.4. Le Retour de Valeur : Quand la Fonction Rend un Résultat....122
- a. L'essentiel de la mission....122
- A. Le mot-clé return : Le messager de la fonction....122

FIGURE 31 : RÉF. IMG (5896-3277-3670)....124

Diagramme d'une fonction qui renvoie un résultat au programme principal....124
- B. Utiliser la valeur retournée....124

FIGURE 32 : RÉF. IMG (4692-3654-3646)....127

Schéma montrant la valeur de retour stockée dans une variable....127
- SCÉNARIOS ET EXERCICES....127

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

3.5. Les Procédures : Fonctions sans Retour de Valeur....129
- a. L'ordre qui ne rapporte rien (en apparence)....129
- A. Définition et Utilité d'une Procédure....129

FIGURE 33 : RÉF. IMG (9855-5516-3035)....131

Schéma d'une procédure : un ordre qui déclenche une action précise....131
- B. Syntaxe et Exemple Pratique....131

FIGURE 34 : RÉF. IMG (1741-5036-0155)....134

Différence visuelle entre une fonction qui retourne une valeur et une procédure....134
- SCÉNARIOS ET EXERCICES....134

3.6. Portée des Variables : Variables Locales et Globales....136
- a. Le secret du chef de quartier....136
- A. La Variable Locale : Le trésor caché dans la fonction....136

FIGURE 35 : RÉF. IMG (6013-9038-0674)....138

Visualisation d'une variable locale confinée dans sa fonction....138
- B. La Variable Globale : L'annonce sur la place du village....138

FIGURE 36 : RÉF. IMG (6523-9520-2115)....141

Schéma d'une variable globale accessible par plusieurs fonctions....141
- b. L'anecdote de la recette de Maman Alphonsine....141

3.7. Les Aiguillages de Sous-Programmes....143
- a. Une logique de carrefour....143
- A. Le Principe de l'Aiguillage....143

FIGURE 37 : RÉF. IMG (5707-4990-2562)....145

Schéma logique d'un aiguillage de sous-programmes depuis un menu principal....145
- B. La Structure SELON...CAS : Le Chef d'Orchestre....145

FIGURE 38 : RÉF. IMG (5283-0775-5611)....148

Organigramme illustrant le fonctionnement de la structure de contrôle SELON...CAS....148
- SCÉNARIOS ET EXERCICES....148

PARTIE 4. LE TRAITEMENT AVANCÉ DES DONNÉES : LES CHAÎNES DE CARACTÈRES....151

Des chiffres au texte....151

La chaîne : une séquence ordonnée....151

Les opérations fondamentales....151

L'utilité pratique au quotidien....151

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

CHAPITRE 4. LA MANIPULATION DE TEXTE **152**

4.1. La Chaîne de Caractères : Un Tableau de Caractères 152

- a. Une séquence de symboles, pas une idée 152
- A. Définition et Structure Interne 152
- B. L'Indexation : La Carte d'Identité de Chaque Caractère 152

FIGURE 39 : RÉF. IMG (2712-8012-1321) **155**

Décomposition d'une chaîne de caractères en tableau indexé 155

- SCÉNARIOS ET EXERCICES 155

4.2. Fonctions de Base : Longueur et Concaténation 157

- a. Compter et assembler le texte 157
- A. La Longueur d'une chaîne : La fonction LEN() 157

FIGURE 40 : RÉF. IMG (7420-2150-9204) **159**

Visualisation du calcul de la longueur d'une chaîne de caractères 159

- B. La Concaténation : L'art d'assembler 159

FIGURE 41 : RÉF. IMG (5885-7648-9552) **161**

Le processus de concaténation pour assembler un nom complet. 161

- SCÉNARIOS ET EXERCICES 161

4.3. Extraction de Sous-chaînes (Gauche, Droite, Milieu) 163

- a. L'art de découper le texte 163
- A. La fonction GAUCHE() 163

FIGURE 42 : RÉF. IMG (2859-3953-4072) **165**

Visualisation des fonctions GAUCHE et DROITE sur le mot "Matadi" 165

- B. La fonction DROITE() 165
- C. La fonction STXT() (ou MILIEU) 166

FIGURE 43 : RÉF. IMG (1858-4008-7769) **168**

Décomposition de la fonction STXT appliquée au mot "Kananga" 168

- b. Scénarios et Exercices 168

4.4. Recherche d'un Caractère ou d'une Sous-chaîne 170

- a. Le jeu de la recherche 170
- A. La Recherche par Caractère Unique 170

FIGURE 44 : RÉF. IMG (8386-5661-9334) **172**

Visualisation de la recherche d'un caractère dans une chaîne de texte 172

- B. La Recherche d'une Sous-chaîne 172

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

FIGURE 45 : RÉF. IMG (6476-7915-8128) **174**

Localisation d'une sous-chaîne dans une phrase de référence congolaise 174

- Scénarios et Exercices 174

4.5. Remplacement et Modification de Chaînes 176

- a. La gomme et le clavier 176
- A. Le remplacement simple : un mot pour un autre 176

FIGURE 46 : RÉF. IMG (7712-1081-8412) **178**

Le mécanisme de la fonction 'Remplacer' dans un traitement de texte 178

- B. Les options de remplacement avancées 178

FIGURE 47 : RÉF. IMG (0503-3091-3480) **181**

Les options avancées du remplacement pour un contrôle total du texte 181

- b. Scénarios et Exercices 181

4.6. Conversion entre Nombres et Chaînes de Caractères 183

- a. Le nombre qui n'en est pas un 183
- A. De la chaîne de caractères au nombre : L'analyse (Parsing) 183

FIGURE 48 : RÉF. IMG (7586-2308-6188) **186**

Schéma de conversion entre le type nombre et le type chaîne 186

- B. Du nombre à la chaîne de caractères : Le formatage 186

4.7. Applications Pratiques du Traitement de Texte 188

- a. L'invisible organisation du texte 188
- A. La Correction et le Remplacement Automatisés 189

FIGURE 49 : RÉF. IMG (6472-4589-0959) **191**

Boîte de dialogue "Rechercher et Remplacer" dans un logiciel de texte 191

- B. L'Extraction et l'Analyse d'Informations 191

FIGURE 50 : RÉF. IMG (4583-5101-3225) **193**

Texte brut à gauche, tableau structuré à droite après traitement 193

- Au-delà du cours : L'histoire du protocole SMS 193

ANNEXES **196**

1. Guide Pédagogique et Concertation 196

- A. Philosophie Pédagogique et Objectifs 196
- B. Séquençage Modulaire et Évaluation 197
- C. La Concertation Pédagogique 198

2. Banque de Projets Intégrateurs 199

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

- A. Cadre Pédagogique et Objectifs.....199
- B. Projets Basés sur le Traitement de Texte (Word).....200
- C. Projets Basés sur le Tableur (Excel).....201
- D. Projets Basés sur la Présentation (PowerPoint).....202

MURHULA BAZIKANGE M.
Directeur Editorial
+243 860 655 590

PRÉLIMINAIRES

1. Finalités et Buts du Cours

1.1. Développement d'une Compétence Instrumentale

Le programme d'études dote l'élève des aptitudes techniques indispensables à une manipulation autonome de l'outil informatique. L'objectif est la maîtrise des logiciels de bureautique fondamentaux, la gestion structurée des fichiers et des dossiers, ainsi que l'utilisation efficiente des périphériques courants. Cette compétence assure une aisance opérationnelle sur un poste de travail standard, préparant l'élève aux exigences académiques et professionnelles futures.

1.2. Structuration d'une Pensée Logique et Analytique

Au-delà de la simple utilisation, le cours vise à construire une approche méthodique des problèmes. L'élève apprend à décomposer une tâche complexe en étapes simples, une compétence fondamentale en algorithmique. Il est initié à l'analyse de l'information numérique, apprenant à distinguer les faits des opinions et à évaluer la fiabilité des sources en ligne. Cette démarche analytique renforce l'esprit critique, une qualité essentielle pour les filières des humanités.

1.3. Instauration d'une Conscience Citoyenne Numérique

L'enseignement de l'informatique intègre une dimension éthique et légale. Le cours sensibilise l'élève aux concepts de l'identité numérique, de la propriété intellectuelle et de la protection des données personnelles. La finalité est de former un citoyen numérique responsable, conscient des risques liés à la cybercriminalité et des bonnes pratiques pour naviguer en toute sécurité. L'élève mesure ainsi l'impact de ses actions dans l'espace public digital.

---

2. Objectif Terminal d'Intégration (OTI)

À l'issue de cette deuxième année des humanités, l'élève mobilise les savoirs, savoir-faire et savoir-être informatiques pour résoudre une situation-problème significative et réaliste. L'OTI évalue la capacité de l'apprenant à intégrer et à utiliser de manière autonome les outils numériques pour produire un document

structuré et pertinent. Cette compétence terminale est l'indicateur principal de la maîtrise des acquis du programme.

2.1. Description de la compétence

La compétence visée amène l'élève à traiter une information brute pour la transformer en un produit numérique fini et communicable. Il s'agit pour lui de planifier, réaliser et finaliser une tâche simple à l'aide des logiciels de bureautique et des principes de base du système d'exploitation. L'évaluation se concentre sur le processus et le résultat, démontrant une autonomie fonctionnelle face à un ordinateur.

2.2. Contexte de la situation d'intégration

L'élève est placé devant une situation-problème issue de son environnement scolaire ou social. Par exemple, il doit rédiger et mettre en forme le rapport d'une activité culturelle organisée au lycée de Mbandaka, ou synthétiser les données d'une petite enquête menée à Bukavu sur les usages du téléphone portable. La situation exige l'utilisation combinée d'un traitement de texte et, éventuellement, d'un tableur pour des calculs simples.

2.3. Productions attendues de l'élève

Concrètement, la performance de l'élève se matérialise par la production d'un ou plusieurs documents numériques qui respectent des consignes précises de forme et de contenu.

Les productions possibles incluent : - Un document texte de plusieurs pages (rapport, exposé) correctement structuré avec titres, paragraphes, et numérotation. - Un tableau simple inséré dans le texte, présentant des données chiffrées. - La sauvegarde du travail sous un nom de fichier explicite, dans une arborescence logique.

---

3. Compétences de Base Attendues

3.1. Maîtrise Instrumentale et Opérationnelle

L'objectif premier est de rendre l'élève autonome sur un poste de travail. Cette compétence inclut la manipulation correcte du système d'exploitation pour la gestion des fichiers, des dossiers et des périphériques. Elle s'étend à l'utilisation efficace des logiciels de bureautique fondamentaux. L'élève doit pouvoir produire un document structuré via un traitement de texte et organiser

des données chiffrées simples dans un tableur, en appliquant les fonctions de base.

3.2. Initiation au Raisonnement Logique et Algorithmique

Parallèlement à l'usage des outils, le cours établit les fondations du raisonnement logique et séquentiel. L'élève est initié à la décomposition d'une tâche simple en une suite d'instructions précises et ordonnées. Cette démarche ne vise pas l'apprentissage d'un langage de programmation, mais la structuration de la pensée. Elle prépare l'élève à la résolution méthodique de problèmes, une aptitude transférable à de nombreuses disciplines des humanités.

3.3. Culture Numérique et Citoyenneté

Enfin, le programme vise à former un citoyen numérique responsable. L'élève est sensibilisé aux enjeux éthiques et sécuritaires de l'environnement digital. Ceci inclut la protection des données personnelles, l'évaluation de la fiabilité des informations en ligne et la compréhension des notions de base du droit d'auteur et de la propriété intellectuelle. La formation promeut un usage critique, légal et éclairé des outils et des ressources numériques.

Partie 1. L'AUTOMATISATION DES TÂCHES RÉPÉTITIVES : LES BOUCLES

La répétition, une punition humaine

L'esprit humain se lasse très vite face à la monotonie. Copier cent fois la même phrase est une punition scolaire classique pour une bonne raison. La fatigue s'installe, la concentration diminue et les erreurs apparaissent inévitablement. L'humain est un être de création, non de répétition mécanique. Cette tâche ingrate est pour lui une source d'épuisement et de frustration.

L'ordinateur, un travailleur infatigable

À l'inverse, un ordinateur ne connaît ni la fatigue ni l'ennui. Il peut exécuter la même instruction un million de fois avec une précision et une vitesse constantes. Calculer les moyennes de tous les élèves d'une école de Mbandaka ou afficher une liste de noms sont des tâches parfaites pour lui. Sa nature même est de suivre des ordres, aussi répétitifs soient-ils.

Donner l'ordre de répéter : la boucle

Pour demander à un ordinateur de répéter une action, le programmeur utilise une structure spéciale. Cette structure porte le nom de « boucle ». Une boucle est une instruction qui dit à la machine : "Exécute ce bloc de code plusieurs fois de suite". Le programmeur précise simplement combien de fois la répétition doit avoir lieu ou sous quelle condition elle doit s'arrêter.

L'enjeu : la puissance et le gain de temps

La maîtrise des boucles donne une puissance de calcul immense. Une tâche qui prendrait des jours à un humain est accomplie en quelques secondes par un programme bien écrit. Le tri des résultats de l'EXETAT pour toute la province du Kasaï illustre bien ce pouvoir. Un programme avec une boucle le fait quasi instantanément, sans aucune erreur de distraction.

Chapitre 1. LES STRUCTURES ITÉRATIVES

1.1. Le Principe de la Répétition Contrôlée

a. L'intelligence de la paresse

Un des faits les plus surprenants en informatique est que les meilleurs programmeurs sont souvent de grands "paresseux intelligents". Ils détestent refaire la même tâche manuellement. Un informaticien ne va jamais taper 100 fois la même ligne de texte. Il préfère passer une heure à écrire un programme qui le fera en une seconde.

Cette idée de faire répéter une action à la machine est au cœur de l'informatique moderne. C'est le principe de l'automatisation. Un ordinateur excelle à répéter une instruction des milliers de fois sans se fatiguer, sans faire d'erreur et à une vitesse inimaginable pour un humain.

A. La boucle : le moteur de la répétition

Une boucle est une structure de code qui exécute un même bloc d'instructions plusieurs fois de suite. Pensez à l'action de piler le manioc pour faire le fufu. Le pilon monte, puis descend. Cette action "monter-descendre" se répète. En programmation, cette action répétée est placée à l'intérieur d'une boucle.

La machine exécute les instructions de la boucle, puis, arrivée à la fin, elle recommence au début de cette même boucle. Cela permet d'automatiser des tâches longues et ennuyeuses. Par exemple, afficher le nom de tous les élèves d'une classe ou calculer la moyenne de 500 étudiants.

B. La condition de sortie : le secret du contrôle

Une boucle ne peut pas tourner à l'infini. Ce serait une catastrophe pour le programme, qui resterait bloqué. Chaque boucle a donc besoin d'une "condition de sortie". C'est une règle qui dit à l'ordinateur quand il doit arrêter de répéter.

Pour notre pilon de fufu, la condition de sortie est : "la pâte est-elle devenue lisse ?". Si la réponse est "oui", on arrête de piler. En informatique, cette condition est souvent basée sur un compteur. Par exemple : "répète 10 fois" ou "répète tant que le compteur est plus petit que 100". Sans cette condition, l'ordinateur continuerait sans jamais s'arrêter.

Figure 1 : Réf. IMG (7221-8963-2594)

Schéma simple d'une boucle contrôlée par une condition de sortie

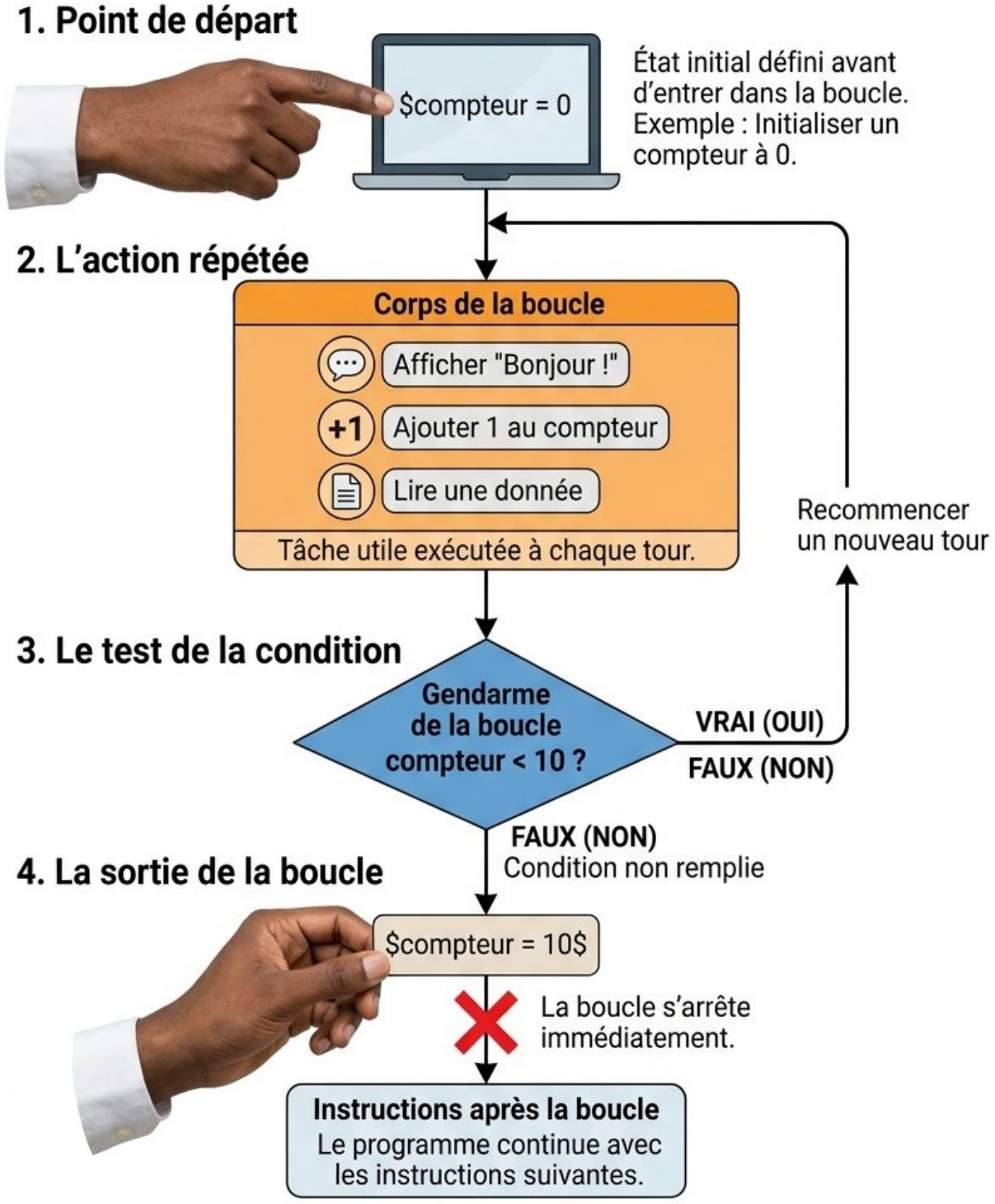


Figure 1 : Réf. IMG (7221-8963-2594)

Schéma simple d'une boucle contrôlée par une condition de sortie

a. Le point de départ

Avant d'entrer dans la boucle, un état initial est défini. C'est par exemple un compteur initialisé à zéro ($compteur=0$). L'ordinateur se prépare à commencer le cycle de répétition. C'est la situation avant que la première action ne soit exécutée.

b. L'action répétée

Ici, l'ordinateur exécute la tâche demandée. Cela peut être "afficher un message", "ajouter 1 au compteur" ou "lire une donnée". Cette partie est le corps de la boucle. C'est le travail utile que la machine doit faire à chaque tour.

c. Le test de la condition

Après chaque exécution, l'ordinateur vérifie la condition de sortie. Par exemple, il se demande : "$compteur<10$?". Si la condition est vraie, il a la permission de recommencer un nouveau tour de boucle. C'est le gendarme de la boucle.

d. La sortie de la boucle

Quand le test de la condition devient faux (par exemple, $compteur$ atteint 10), la boucle s'arrête immédiatement. Le programme ne recommence plus le cycle et continue son exécution avec les instructions qui se trouvent après la boucle. Le processus est terminé.

■ ■ ■

C. Types de boucles courantes

Il existe plusieurs manières de construire des boucles. Les deux plus célèbres sont la boucle "TANT QUE" (while) et la boucle "POUR" (for).

- **La boucle TANT QUE** : Elle répète les instructions tant qu'une condition est vraie. On l'utilise quand on ne sait pas exactement combien de fois la boucle doit tourner. Exemple : "Tant qu'il y a des clients dans la file, sers le prochain client".

- **La boucle POUR** : Elle est parfaite pour répéter une action un nombre de fois bien défini. Exemple : "Pour chaque élève de 1 à 45, fais l'appel".

Le choix entre ces deux types de boucles dépend uniquement du problème à résoudre.

SCÉNARIOS ET EXERCICES

L'anecdote : Le décompte du gardien de nuit

Papa Boloko est gardien de nuit pour un grand entrepôt à Kisangani, dans la Province Orientale. Son patron lui demande de faire une ronde toutes les heures et de vérifier que les 15 portes de l'entrepôt sont bien fermées. Au début, Papa Boloko se perdait dans son décompte. Il oubliait s'il en était à la porte 12 ou 13.

Un jour, son neveu, élève en informatique, lui a expliqué le principe de la boucle. Il lui a conseillé de prendre 15 petits cailloux. À chaque porte vérifiée, il déplace un caillou d'une poche à l'autre. L'action est "vérifier une porte". La boucle se répète "tant qu'il reste des cailloux dans la première poche". La condition de sortie est claire : quand la première poche est vide, la ronde est terminée. Papa Boloko a automatisé son travail sans ordinateur !

Exercice 1 : Les mangues de Tante Hélène

Énoncé : Tante Hélène, qui vend des fruits au marché de Mbandaka en Équateur, veut préparer des paniers de 12 mangues. Décris le processus qu'elle suit en utilisant la logique d'une boucle.

Résolution pas-à-pas :

1. **Étape 1 : Initialisation** Tante Hélène prend un panier vide. Le nombre de mangues dans le panier est de 0.
2. **Étape 2 : Définition de la boucle et de son action** L'action à répéter est : "mettre une mangue dans le panier".
3. **Étape 3 : Définition de la condition de sortie** La boucle doit continuer "tant que le nombre de mangues dans le panier est inférieur à 12". Dès que le panier contient 12 mangues, le processus s'arrête pour ce panier.

Exercice 2 : L'appel des élèves à Kikwit

Énoncé : Un enseignant de l'Institut Nzolana à Kikwit, dans le Kwilu, doit faire l'appel pour sa classe de 45 élèves. Comment ce processus peut-il être vu comme une boucle de type "POUR" ?

Résolution pas-à-pas :

1. **Étape 1 : Identifier l'action répétitive** L'action principale que l'enseignant répète est : "lire le nom d'un élève sur la liste et marquer sa présence ou son absence".
2. **Étape 2 : Déterminer la structure de la boucle "POUR"** C'est une boucle "POUR" car le nombre de répétitions est connu à l'avance : 45. On peut la formuler ainsi : "POUR chaque élève allant de 1 à 45, faire l'appel".
3. **Étape 3 : Conclure sur l'arrêt du processus** La boucle démarre avec le premier élève et s'arrête automatiquement après que le 45ème élève a été appelé. L'enseignant sait qu'il a terminé quand il arrive à la fin de sa liste.

1.2. La Boucle à Comptage « Pour… De… À… »

a. Une tâche prévisible

Pendant la révolution industrielle, les usines ont systématisé la production. Un ouvrier répétait la même action des centaines de fois par jour : visser un boulon, fixer une pièce, emballer un produit. Cette répétition, bien que fatigante pour l'humain, garantissait une production rapide et uniforme. Le nombre de répétitions était connu à l'avance : 500 voitures par jour signifiaient 500 volants à installer.

L'informatique s'inspire de cette logique. Pour des tâches dont le nombre d'exécutions est fixé, une structure de contrôle spécifique existe. Elle agit comme un chef d'équipe qui connaît exactement le nombre de pièces à produire avant de commencer le travail.

A. Définition et Syntaxe

La boucle « Pour » exécute un bloc d'instructions un nombre de fois précis et connu à l'avance. Elle utilise une variable spéciale, le compteur, pour suivre le nombre de tours effectués.

Sa syntaxe générale en algorithmique est la suivante :

POUR compteur **DE** valeur_initiale **À** valeur_finale **[PAS de** pas**]** < *Liste des instructions à répéter* > **FIN POUR**

Les éléments de cette structure sont : - **compteur** : Une variable qui stocke le tour actuel de la boucle. - **valeur_initiale** : Le nombre auquel le compteur commence. - **valeur_finale** : Le nombre auquel le compteur s'arrête. - **PAS** : Optionnel. La valeur d'incrémentation (ou décrémentation) du compteur à chaque tour. Si ce mot-clé est absent, le pas est de 1 par défaut.

B. Le Mécanisme du Compteur

L'ordinateur suit un processus rigoureux en quatre temps pour gérer une boucle POUR.

Figure 2 : Réf. IMG (2810-7545-2744)

Schéma du fonctionnement d'une boucle à comptage avec son compteur.

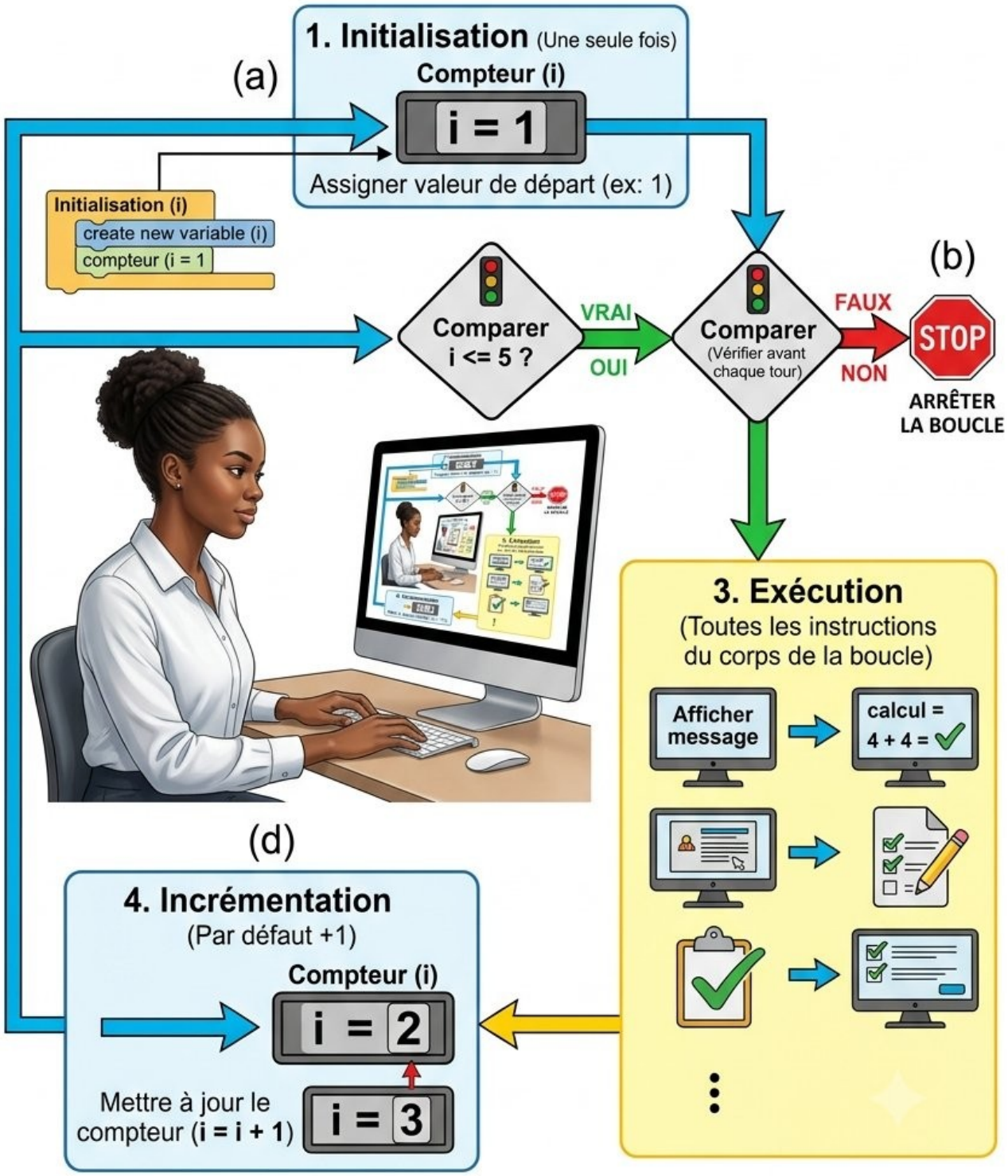


Figure 2 : Réf. IMG (2810-7545-2744)

Schéma du fonctionnement d'une boucle à comptage avec son compteur.

a. L'initialisation

Au tout début, l'ordinateur crée une variable, le compteur. Il lui assigne la valeur de départ. Par exemple, si la boucle est POUR i DE 1 À 5, le compteur i reçoit la valeur 1. Cette étape ne se produit qu'une seule fois.

b. La condition de continuation

Avant chaque tour, l'ordinateur compare la valeur du compteur à la valeur finale. Il vérifie si i est inférieur ou égal à 5. Si la condition est vraie (1 <= 5, 2 <= 5…), le corps de la boucle est exécuté. Si elle est fausse, la boucle s'arrête.

c. L'exécution du bloc

Si la condition est validée, toutes les instructions entre POUR et FIN POUR sont exécutées. L'ordinateur fait ce qui lui est demandé : afficher un message, faire un calcul. C'est le cœur du travail répétitif.

d. L'incrémentation

Juste après l'exécution, le compteur est mis à jour. Par défaut, il augmente de 1. Notre i passe de 1 à 2, puis de 2 à 3, et ainsi de suite. Le cycle recommence alors à l'étape de la condition.

b. L'appel des finalistes au TENASOSP

À Mbandaka, le chef du centre de correction de l'examen TENASOSP doit afficher la liste des 10 premiers lauréats de la province de l'Équateur. Au lieu d'écrire 10 fois « Appeler le candidat numéro… », il imagine un petit programme. Il sait que la tâche doit être répétée exactement 10 fois, en commençant par le numéro 1 et en finissant par le numéro 10. La boucle POUR est l'outil parfait pour cette mission. Elle automatise l'appel, évite les erreurs d'oubli et garantit que chaque lauréat, de 1 à 10, est bien nommé. C'est une application directe d'une boucle à comptage dans une situation administrative concrète.

Exercice 1 : La table de multiplication du Kwilu

Énoncé : Papa Mavinga, enseignant à Bandundu-Ville, veut créer un programme qui affiche les 12 premiers résultats de la table de multiplication de 7. Aidez-le à écrire l'algorithme.

Résolution pas-à-pas :

1. **Étape 1 : Identifier le besoin** La tâche consiste à afficher le résultat de 7 x 1, 7 x 2, jusqu'à 7 x 12. La répétition se fait 12 fois. Le nombre de tours est connu à l'avance, donc une boucle POUR est parfaitement adaptée.

2. **Étape 2 : Définir les paramètres de la boucle** Le compteur doit commencer à 1 et s'arrêter à 12. Le pas par défaut (1) est suffisant. Ce compteur représentera le nombre par lequel 7 est multiplié à chaque tour.

3. **Étape 3 : Écrire l'algorithme** L'algorithme utilise une variable i comme compteur et une variable resultat pour stocker le calcul.

```
ALGORITHME Table_De_7

VARIABLES
   i : ENTIER
   resultat : ENTIER

DEBUT
   POUR i DE 1 À 12
      resultat ← 7 * i
      AFFICHER "7 x ", i, " = ", resultat
   FIN POUR
FIN
```

Exercice 2 : Compte à rebours pour le maïs du Kasaï

Énoncé : Maman Tshiela, à Kananga, met du maïs à bouillir. La cuisson dure 10 minutes. Elle veut un programme qui affiche un compte à rebours de 10 à 1, puis affiche « Le maïs est prêt ! ».

Résolution pas-à-pas :

1. **Étape 1 : Analyser la répétition** Il faut compter de 10 jusqu'à 1. Le nombre de répétitions est connu (10 fois), mais la direction est inversée (décroissante).

2. **Étape 2 : Utiliser le mot-clé PAS** Pour un compte à rebours, le compteur doit diminuer. Il faut donc utiliser le mot-clé PAS avec une valeur négative. Ici, le pas sera de -1. La valeur initiale est 10 et la valeur finale est 1.

3. **Étape 3 : Construire l'algorithme** Le message final « Le maïs est prêt ! » doit être affiché une seule fois, après la fin de la boucle.

```
ALGORITHME Compte_Rebours_Mais

VARIABLES
   minute : ENTIER

DEBUT
   POUR minute DE 10 À 1 PAS -1
      AFFICHER "Temps restant : ", minute, " minute(s)"
   FIN POUR

   AFFICHER "Le maïs est prêt !"
FIN
```

1.3. Anatomie d'une Boucle : Initialisation, Condition, Incrémentation

a. Le secret d'une tâche bien répétée

Même si un ordinateur semble exécuter des milliers de tâches complexes, son intelligence repose souvent sur sa capacité à répéter une action simple avec une discipline de fer. Répéter n'est pas synonyme de chaos. Au contraire, chaque répétition est parfaitement contrôlée, comme un soldat qui compte ses pas.

Pour qu'une machine répète une tâche sans se tromper ni tourner à l'infini, elle a besoin d'un plan en trois étapes. Ce plan est la véritable anatomie d'une boucle. Il lui donne un point de départ, une raison de continuer et une instruction pour avancer vers la fin.

A. L'Initialisation : Le Point de Départ

Toute répétition commence quelque part. L'initialisation est l'acte de préparer la boucle en donnant une valeur de départ à une variable. Cette variable est

souvent appelée un "compteur". Elle sert de référence, comme le premier élève d'une rangée ou le kilomètre zéro sur une route qui part de Matadi.

En pseudo-code, cette étape ressemble à ceci : compteur PREND_LA_VALEUR 1

Cette instruction prépare le terrain. Le compteur est maintenant prêt, sa valeur initiale est fixée à 1. La boucle peut se préparer à démarrer.

B. La Condition : Le Gardien de la Boucle

Une boucle ne peut pas tourner éternellement. La condition est une question posée à chaque tour pour savoir si la boucle doit continuer ou s'arrêter. C'est le gardien qui vérifie si vous avez encore le droit de passer. Tant que la réponse à la question est "VRAI", la boucle exécute son code. Dès que la réponse devient "FAUX", la boucle s'arrête net.

Par exemple, la condition peut être : compteur EST_INFERIEUR_OU_EGAL_A 10

À chaque tour, l'ordinateur vérifie la valeur du compteur. Si compteur vaut 5, la condition est vraie. Si compteur vaut 11, elle devient fausse et la porte se ferme.

Figure 3 : Réf. IMG (0347-2776-7627)

Les trois piliers d'une boucle : Départ, Test et Progression

a. Le Départ (Initialisation)



1. INITIALISATION DE LA VARIABLE

Variable initialisée

Compteur = 1

b. Le Test (Condition)



2. TEST DE LA CONDITION

Condition de sortie

COMPTEUR < CIBLE ? (Vrai)

→ Continuer la course

COMPTEUR = CIBLE ? (Faux)

→ Terminer la course

(e.g., Compteur < 5)

c. La Progression (Incrémentation)



3. INCRÉMENTATION DE LA VARIABLE

Pas en avant (Incrément)

Compteur = Compteur + 1

d. Le Cycle Complet

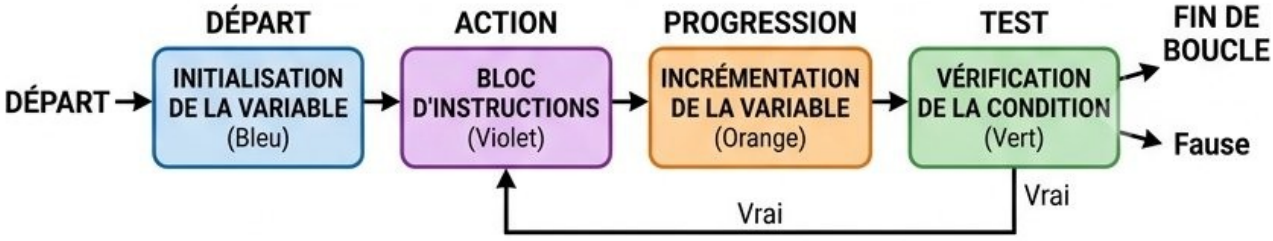


Figure 3 : Réf. IMG (0347-2776-7627)

Les trois piliers d'une boucle : Départ, Test et Progression

a. Le Départ (Initialisation)

Le point de départ est représenté par une ligne de départ sur une piste de course. C'est là que le coureur (la boucle) se positionne. La variable est initialisée, par exemple à 0 ou 1, avant que la course ne commence.

b. Le Test (Condition)

Un arbitre se tient à chaque tour de piste. Il vérifie si le coureur a atteint le nombre de tours requis. Tant que le nombre de tours est inférieur à la cible, l'arbitre fait signe de continuer. C'est la condition de la boucle.

c. La Progression (Incrémentation)

Chaque fois que le coureur termine un tour, son compteur de tours augmente de 1. Ce pas en avant est l'incrémentation. Sans cette progression, le coureur tournerait en rond pour l'éternité, car la condition de l'arbitre ne serait jamais satisfaite.

d. Le Cycle Complet

L'ensemble forme un circuit fermé : Départ -> Action -> Progression -> Test. Ce cycle se répète jusqu'à ce que le test de l'arbitre (la condition) échoue, mettant fin à la course (la boucle).

C. L'Incrémentation : Le Moteur du Progrès

Sans progression, une boucle est un piège infini. L'incrémentation est l'opération qui modifie la variable compteur à la fin de chaque tour. Le plus souvent, on ajoute 1 à sa valeur. Cette modification est cruciale. Elle assure que la variable se rapproche progressivement de la valeur qui rendra la condition fausse. C'est le moteur qui fait avancer la boucle vers sa conclusion.

L'instruction d'incrémentation typique est : compteur PREND_LA_VALEUR compteur + 1

Cette ligne garantit que le compteur passera de 1 à 2, puis de 2 à 3, et ainsi de suite, jusqu'à ce qu'il atteigne la limite fixée par la condition.

Scénarios et Exercices

L'anecdote du grand marché de Goma

Au marché Virunga de Goma, Maman Sylvie vend des pagnes wax hollandais. Chaque matin, pour son inventaire, elle doit compter les 25 nouveaux pagnes qu'elle vient de recevoir du fournisseur du Kasaï. Au début, elle se perdait souvent dans son comptage. Un jour, son fils, qui apprend l'informatique, lui explique le principe de la boucle. "Maman, c'est simple. 1. Tu commences par le premier pagne (initialisation à 1). 2. Tu comptes un pagne, puis tu le mets de côté. Tu avances ton compte de 1 (incrémentation). 3. Tu te poses la question : ai-je déjà compté 25 pagnes ? (condition). Si la réponse est non, tu recommences l'étape 2. Si c'est oui, tu as terminé." Maman Sylvie applique cette méthode et ne se trompe plus jamais. Elle a, sans le savoir, appliqué une boucle algorithmique parfaite.

Exercice 1 : Le décompte des fûts d'huile de palme

Énoncé : Papa Bondo, un transporteur fluvial de Mbandaka, doit charger 5 fûts d'huile de palme sur sa pirogue. Aidez-le à créer un programme simple qui affiche à l'écran "Chargement du fût numéro 1", "Chargement du fût numéro 2", etc., jusqu'au cinquième fût.

Résolution pas-à-pas :

1. **Étape 1 : Initialisation** Nous avons besoin d'un compteur pour suivre les fûts. Appelons-le fut_actuel. Sa valeur de départ est 1. fut_actuel PREND_LA_VALEUR 1
2. **Étape 2 : Condition** Le chargement doit s'arrêter après le 5ème fût. La boucle doit donc continuer tant que fut_actuel est inférieur ou égal à 5. TANT_QUE fut_actuel <= 5
3. **Étape 3 : Incrémentation** Après avoir affiché le message pour un fût, nous devons passer au suivant. Il faut donc ajouter 1 à fut_actuel à chaque tour de boucle. fut_actuel PREND_LA_VALEUR fut_actuel + 1
4. **Étape 4 : Assemblage de la boucle** En combinant ces éléments, nous obtenons une boucle POUR qui gère tout cela automatiquement. POUR fut_actuel ALLANT DE 1 A 5 : AFFICHER "Chargement du fût numéro ", fut_actuel

Exercice 2 : Afficher les tables de multiplication

Énoncé : Faida, une élève de l'école Tushinde à Lubumbashi, veut utiliser l'informatique pour réviser ses tables de multiplication. Créez un programme qui affiche la table de 3, de 3 x 1 = 3 jusqu'à 3 x 10 = 30.

Résolution pas-à-pas :

1. **Étape 1 : Initialisation du multiplicateur** Nous avons besoin d'une variable qui va changer de 1 à 10. Appelons-la multiplicateur. multiplicateur PREND_LA_VALEUR 1
2. **Étape 2 : Définir la condition** La table s'arrête à 10. La boucle doit continuer tant que multiplicateur est inférieur ou égal à 10. TANT_QUE multiplicateur <= 10
3. **Étape 3 : Calculer et afficher le résultat** Dans la boucle, nous devons calculer le produit et l'afficher dans un format clair. resultat PREND_LA_VALEUR 3 * multiplicateur AFFICHER "3 x ", multiplicateur, " = ", resultat
4. **Étape 4 : Incrémenter le multiplicateur** Pour passer de 3x1 à 3x2, puis 3x3, etc., il faut augmenter le multiplicateur de 1 à chaque tour. multiplicateur PREND_LA_VALEUR multiplicateur + 1

1.4. La Boucle Conditionnelle « Tant que... Faire... »

a. Le principe de la vérification préalable

Lorsque l'on puise de l'eau à une pompe manuelle dans un village du Kwilu, on vérifie d'abord si le seau est plein. Si le seau n'est pas plein, on actionne la pompe. On répète cette action. On vérifie à nouveau. Dès que le seau est plein, on arrête de pomper.

Cette logique de "vérifier avant d'agir" est le cœur de la boucle "Tant que". C'est une structure qui exécute une série d'instructions de manière répétée, mais seulement si une condition spécifique est vraie au départ. Si la condition est fausse dès le début, aucune action n'est effectuée.