

Sommaire

1. Introduction à l'informatique et à la programmation

Qu'est-ce que la programmation

Algorithmique vs codage

Organisation d'un programme

2. Notions de base : types, variables, opérateurs

Types de données entiers, réels, booléens, chaînes...

Variables, affectation, constantes

Opérateurs arithmétiques, logiques, relationnels

3. Structures de contrôle

Instructions conditionnelles `if`, `switch` selon le langage

Boucles et itérations `for`, `while`, `do-while`...

Séquences d'instructions, portée des variables

4. Fonctions / Procédures / Sous-programmes

Définition et appel de fonctions/subroutines

Passage d'arguments et retour de résultat

Modularisation du code

5. Tableaux et structures de données simples

Tableaux unidimensionnels, multidimensionnels

Manipulation d'éléments, parcours, remplissage

Utilité pour le traitement de données

6. Entrées / Sorties (I/O) et gestion des données

Lecture de l'utilisateur, écriture à l'écran/fichier

Formats d'affichage, saisie, validation

Fichiers (si le livre couvre l'accès aux fichiers)

7. Algorithmique : résolution de problèmes

Analyse de problèmes et formalisation

Conception d'algorithmes par étapes, pseudo-code

Traduction en code

8. Exercices progressifs commentés et corrigés

Du simple calcul, conditions au plus complexe tableaux, boucles imbriquées

Études de cas pratiques

Correction + explication des choix de solution

9. Approfondissements / Thèmes avancés optionnels selon l'édition

Recursion

Structures de données avancées listes, etc.

Notions de base de la programmation modulaire

10. Annexes / Notions complémentaires

Rappels d'algorithmique

Notions de base sur les erreurs, débogage

Conseils de bonnes pratiques de programmation