Sommaire :

---

1. Introduction à l'informatique

- Définitions : algorithme, programme, machine
- Histoire et objectifs de l'informatique
- Présentation du langage OCaml

---

2. Types et instructions de base

- Types simples : entiers, booléens, caractères
- Instructions : condition, séquence, itération
- Fonctions simples et récursives
- Entrées et sorties

---

3. Listes et récursivité

- Définition et création de listes
- Parcours récursif des listes
- Fonctions classiques sur les listes (`map`, `filter`, `fold`)
- Listes infinies et paresseuses (introduction)

---

4. Structures linéaires

- Piles et files : définitions, applications
- Implémentations par listes
- Exercices d'application

---

5. Arbres

- Arbres binaires
- Parcours : préfixé, infixé, postfixé
- Représentation et manipulation en OCaml
- Arbres d'expressions

---

6. Programmation modulaire

- Modules en OCaml
- Types abstraits
- Conception modulaire d'un programme
- Interfaces et encapsulation

---

7. Algorithmes fondamentaux

- Recherche : linéaire, dichotomique
- Tri : insertion, fusion
- Notion de complexité en temps
- Analyse qualitative des performances

---

8. Logique booléenne

- Opérateurs logiques
- Équivalences et simplifications
- Tables de vérité
- Circuits logiques combinatoires simples
- Codage de fonctions logiques en OCaml

---

9. Types algébriques et programmation avancée

- Types produits, sommes, récursifs
- Utilisation de `match` et `option`
- Traitement d'erreurs fonctionnel
- Exemple de mini-interpréteurs

---

10. Langages formels (notions)

- Alphabets, mots, langages
- Définition formelle d'un langage
- Premiers pas vers les expressions régulières

---

11. Théorie du calcul (initiation)

- Qu'est-ce qu'un algorithme « universel » ?
- Machine de Turing (introduction)
- Limites de l'algorithmique

Annexes

- Aide-mémoire OCaml (syntaxe)
- Index des fonctions utilisées
- Exercices d’approfondissement avec corrigés