

# NFA037

## Progression pédagogique

Document non contractuel

13 séances hebdomadaires de 3,5 heures, exclusivement sur machines.  
Le rythme est soutenu car le programme est vaste.

Lors de chaque séance : les notions essentielles de cours sont présentées et illustrées d'exemples concrets de codes source compilés et exécutés portant sur le thème. De nombreux exercices seront proposés et corrigés sur chaque thème.

Les auditeurs programment, en permanence sur machine, des exercices qui leur sont proposés. Au cours de la séance, l'enseignant intervient pour rappeler certains points essentiels de cours, ajouter des points importants, apporter des précisions, corriger certains exercices ou répondre aux questions.

Le code source solution des exercices est fourni. Les programmes sont implémentés, en particulier avec les listes et fichiers texte.

Un examen terminal sur feuille de 2 heures minimum reprenant l'ensemble des notions vues en séances (en particulier les fonctions, la récursivité, les pointeurs et les listes chaînées) est proposé en fin de semestre.

Les auditeurs devront impérativement écrire des lignes de code source en C ou C++.

## Conseils aux auditeurs

Il est conseillé aux auditeurs de travailler (chercher et écrire des codes sources devant la machine) au moins 2 fois 2 heures par semaine à la maison en plus de la séance hebdomadaire. Copier-coller ou lire des codes sources fonctionnant n'a aucun intérêt pédagogique. C'est en cherchant et en corrigeant ses erreurs que l'on progresse en programmation. Et cela prend du temps.

## Déroulé des séances

### Séance 1

Installation de Codeblocks ; entrées/sorties / variables.

### Séance 2

Tests conditionnels if / switch. Imbrications.

### Séance 3

Boucles. Imbrication de boucles. Présentation des tableaux.

### Séance 4

Les tableaux. Tableaux à 2 dimensions. Tableaux dynamiques. Petits algorithmes.

### Séance 5

Fonctions. Passage de paramètres par valeur.

### Séance 6

Récursivité.

**Séance 7**

Fin des fonctions récursives. Pointeurs. Passage de paramètres par adresse.  
Gestion de la mémoire.

**Séance 8**

Types complexes. Chaînes de caractères.

**Séance 9**

Lecture et écriture dans des fichiers textes. Petits algorithmes.

**Séance 10**

Fin des fichiers texte. Listes chaînées (création, ajout/suppression de maillons...).  
Algorithmes sur les listes.

**Séance 11**

Fin des listes chaînées. De C à C++. Programmes en C++. Surcharges.

**Séance 12**

Programmation orientée objet en C++ (classes, variables d'instance, méthodes, objets). Implémentation de classes. Encapsulation de données. Héritage en C++.

**Séance 13**

Héritage en C++. Algorithmes.

Document non contractuel donné à titre indicatif