



Essentiel d'informatique

2023/2024

Victor Sarrazin



Bienvenue dans l'essentiel d'informatique de mes cours de prépa. Ce document a pour objectif de contenir l'intégralité des cours d'informatique afin de les condenser et de les adapter.

Bonne lecture...

Sommaire

Introduction aux langages :

 I Introduction au C	3
 II Introduction au OCaml	5

Structures de données :

 I Structures de données	6
 II Piles, files, dictionnaires	6
 III Arbres	6
 IV Graphes	6

Informatique théorique :

 I Bases	7
 II Récursion	7
 III Stratégies algorithmiques	7

Introduction

I Introduction au C

I.1 Variables

Pour définir une variable en C on a la syntaxe suivante : type nom

```
1 int mango = 0;
```



Il est possible de définir plusieurs variables en même temps :

```
1 int banana = apple = 12;
```



I.2 Opérateurs

On a les opérations arithmétiques suivantes :

Opération	En C
Addition	a + b
Soustraction	a - b
Multiplication	a * b
Division	a / b
Modulo	a % b

On peut utiliser +=, -=, *=, /= et %= pour faire des opérations arithmétiques et des assignments

De plus on peut utiliser ++ et -- pour incrémenter/décrémenter

Les comparaisons se font avec >, >=, <=, < et ==.

On a des opérateurs binaires && (et logique), || (ou logique) et ! (négation de l'expression suivante)



Attention :

Le && est prioritaire sur le ||

I.3 Structures de contrôle

Pour exécuter de manière conditionnelle, on utilise if (cond) {...} else if (...) {} ...
{ } else { }

Ainsi le code suivant est valide :

```

1  if (x == 1) {
2      // Do code
3  } else if (x > 12) {
4      // Do code bis
5  } else {
6      // Do code ter
7  }

```



Attention :

En C un 0 est considéré comme false et toute autre valeur numérique true

Pour faire une boucle on peut utiliser un while (cond) {} qui exécute le code tant que la condition est valide

On peut utiliser do {} while (cond) qui exécute une fois puis tant que la condition est vérifiée

Il est aussi possible d'utiliser for (...) {}, de la manière suivante :

```

1  // De 0 à n - 1
2  for (int i = 0; i < n; i++) {
3
4  }
5
6  // De 0 à n - 1 tant que cond
7  for (int i = 0; i < n && cond; i++) {
8
9  }

```

A noter qu'en C il est possible de modifier la valeur de i et donc de sortir plus tôt de la boucle

Il est possible de sortir d'une boucle avec break, ou de passer à l'itération suivante avec continue



I.4 Fonctions

Pour définir une fonction on écrit :

```

1  int my_func(int a, int b) {
2      // Do code
3      return 1;
4  }

```

Si on ne prend pas d'arguments on écrit int my_func(void) {} et si on ne veut rien renvoyer on utilise void my_func(...) {}

Ainsi pour appeler une fonction on fait :

```

1  int resp = my_func(12, 14);

```

On peut déclarer une fonction avant de donner son code mais juste sa signature avec :

```

1  int my_func(int);

```

I.5 Tableaux en C

Le type d'un tableau en C est `type[]` ou `* type`

Pour initialiser un tableau on a les manières suivantes :

```
1  int[4] test = {0, 1, 2, 3}; // Initialise un tableau de taille 4 avec 0,1,2,3
2  int[] test = {0, 1, 2, 3}; // Initialise un tableau avec 0,1,2,3 (avec 4
3  éléments)
   int[4] test = {0}; // Initialise un tableau de taille 4 avec 0,0,0,0
```



Il n'est pas obligé de donner la taille d'un tableau elle sera déterminée au moment de l'exécution



Attention :

Si on dépasse du tableau C ne prévient pas mais s'autorise à faire n'importe quoi

Pour affecter dans une case de tableau on fait :

```
1  test[1] = test[2] // On met dans la case 1 la valeur de la case 2
```



II Introduction au OCaml

Structures de données

 I Structures de données

 II Piles, files, dictionnaires

 III Arbres

 IV Graphes

Informatique théorique

I Bases

I.1 Fonctions

On dit qu'une fonction a des **effets de bord** si son exécution a des conséquences sur d'autres choses que ses variables locales

Une fonction est **déterministe** si le résultat est toujours le même avec les mêmes arguments

















Une fonction est dite **pure** lorsqu'elle est déterministe et sans effets de bord

I.2 Algorithmes de tri

II Récursion

III Stratégies algorithmiques

Table des matières

 I Introduction au C	3
 I.1 Variables	3
 I.2 Opérateurs	3
 I.3 Structures de contrôle	3
 I.4 Fonctions	4
 I.5 Tableaux en C	5
 II Introduction au OCaml	5
 I Structures de données	6
 II Piles, files, dictionnaires	6
 III Arbres	6
 IV Graphes	6
 I Bases	7
 I.1 Fonctions	7
 I.2 Algorithmes de tri	7
 II Récursion	7
 III Stratégies algorithmiques	7
Table des matières	8