

# Chapitre 1 : Introduction

Informatique de base  
2013-2014

Sup Galilée

## Source

Ce cours est très largement inspiré du polycopié de cours de [Jacques Le Maître](#), professeur émérite d'informatique à l'Université du Sud Toulon-Var.

# Objectif du cours

L'objectif de ce cours est d'apprendre à programmer en [langage C](#).

Il s'agit donc non seulement de savoir écrire un programme, mais également de comprendre de quoi un programme est constitué, et ce qu'il réalise.

Un programme est essentiellement constitué d'expressions et d'instructions, notions que nous allons étudier dans ce cours.

# Le langage C, son origine et son importance

Le langage C a été créé au début des années 1970 par des chercheurs du laboratoire Bell aux États-Unis.

Il a été conçu, à l'origine, pour être le langage de développement du système d'exploitation UNIX.

Depuis sa création, ce langage a été très largement diffusé. Cela explique le choix de ce langage comme support d'un cours introductif à la programmation.

Le langage C est à la base d'autres langages plus modernes tels que le langage C++ et Java.

## Exemple de programme C

Nous allons commencer ce cours par un exemple de programme en langage C.

Nous allons voir, *pas-à-pas*, ce que réalise ce programme, c'est-à-dire, comprendre l'*exécution* de ce programme.

Vous observerez que les numéros de ligne du programme sont indiqués mais ils ne font pas partie du programme lui-même. Ils ne servent qu'à repérer les différentes parties (les *instructions*) du programme.

```
(01) #include <stdio.h>
(02) /* Maximum de deux nombres entiers */
(03) main ()
(04) {
(05)     int x, y, max;
(06)     printf("x ? ");
(07)     scanf("%d",&x);
(08)     printf("y ? ");
(09)     scanf("%d",&y);
(10)     if (x >= y)
(11)         max = x;
(12)     else
(13)         max = y;
(14)     printf("max = %d", max);
(15) }
```

## Ligne numéro 1

```
(01) #include <stdio.h>
(02) /* Maximum de deux nombres entiers */
(03) main ()
(04) {
(05)     int x, y, max;
(06)     printf("x ? ");
(07)     scanf("%d",&x);
(08)     printf("y ? ");
(09)     scanf("%d",&y);
(10)     if (x >= y)
(11)         max = x;
(12)     else
(13)         max = y;
(14)     printf("max = %d", max);
(15) }
```

## Ligne numéro 1

La **ligne 1** indique l'utilisation de la librairie **stdio.h** des entrées et sorties standard.

Cette librairie contient notamment les définitions des "fonctions" permettant d'afficher ou de saisir au clavier des valeurs (caractères ou nombres).

La fonction permettant l'affichage est **printf** (on la retrouve aux lignes 6, 8 et 14), et celle permettant la saisie de valeurs est **scanf** (lignes 7 et 9).

Sans la ligne **#include <stdio.h>**, il est impossible d'utiliser **printf** et **scanf**.



## Ligne numéro 2

```
(01) #include <stdio.h>
(02) /* Maximum de deux nombres entiers */
(03) main ()
(04) {
(05)     int x, y, max;
(06)     printf("x ? ");
(07)     scanf("%d",&x);
(08)     printf("y ? ");
(09)     scanf("%d",&y);
(10)     if (x >= y)
(11)         max = x;
(12)     else
(13)         max = y;
(14)     printf("max = %d", max);
(15) }
```

## Ligne numéro 2

La **ligne 2** est un **commentaire**, c'est-à-dire une suite de caractères qui n'est pas prise en compte dans l'exécution d'un programme.

En général, un commentaire permet d'expliquer ce que réalise une partie d'un programme.

Dans notre exemple, il indique ce que calcule le programme.

## Lignes 3 à 15

```
(01) #include <stdio.h>
(02) /* Maximum de deux nombres entiers */
(03) main ()
(04) {
(05)     int x, y, max;
(06)     printf("x ? ");
(07)     scanf("%d",&x);
(08)     printf("y ? ");
(09)     scanf("%d",&y);
(10)     if (x >= y)
(11)         max = x;
(12)     else
(13)         max = y;
(14)     printf("max = %d", max);
(15) }
```

## Lignes 3 à 15

Les **lignes 3 à 15** constituent la **définition** de la fonction **main** (ou **fonction principale**).

Un programme C est constitué d'une séquence de définitions de fonctions. Parmi elles, la fonction **main** doit toujours être présente et est exécutée la première (d'où son nom).

Le **corps** de cette fonction est le **bloc** qui commence à la **ligne 4** et se termine à la **ligne 15**.

## Ligne 5

```
(01) #include <stdio.h>
(02) /* Maximum de deux nombres entiers */
(03) main ()
(04) {
(05)     int x, y, max;
(06)     printf("x ? ");
(07)     scanf("%d",&x);
(08)     printf("y ? ");
(09)     scanf("%d",&y);
(10)     if (x >= y)
(11)         max = x;
(12)     else
(13)         max = y;
(14)     printf("max = %d", max);
(15) }
```

## Ligne 5

La **ligne 5** est la **déclaration** de trois **variables** x, y, et max de **type** int (pour “integer”, nombre entier).

À la suite de cette déclaration trois **cases**, désignées par x, y, et max, sont allouées dans la mémoire du programme.

Ces cases contiendront chacune un **nombre entier** (une donnée du type des variables).

## Lignes 6 et 7

```
(01) #include <stdio.h>
(02) /* Maximum de deux nombres entiers */
(03) main ()
(04) {
(05)     int x, y, max;
(06)     printf("x ? ");
(07)     scanf("%d",&x);
(08)     printf("y ? ");
(09)     scanf("%d",&y);
(10)     if (x >= y)
(11)         max = x;
(12)     else
(13)         max = y;
(14)     printf("max = %d", max);
(15) }
```

## Lignes 6 et 7

La **ligne 6** est une instruction qui **affiche à l'écran** le message :

**x ?**

La **ligne 7** est une instruction qui permet la **saisie** d'une valeur entière qui sera attribuée à la variable **x**.

C'est la lettre **d**, située après le signe pourcentage (**%**), qui indique que l'on doit saisir un entier.

Puis **&x** signifie que la variable **x** recevra comme valeur l'entier saisi au clavier (on lui **affecte** cette valeur).

Les lignes 8 et 9 agissent de même pour la variable **y**.

Rappelons que l'on peut utiliser ces fonctions puisqu'on inclut la librairie **stdio**.



## Lignes 10 à 13

```
(01) #include <stdio.h>
(02) /* Maximum de deux nombres entiers */
(03) main ()
(04) {
(05)     int x, y, max;
(06)     printf("x ? ");
(07)     scanf("%d",&x);
(08)     printf("y ? ");
(09)     scanf("%d",&y);
(10)     if (x >= y)
(11)         max = x;
(12)     else
(13)         max = y;
(14)     printf("max = %d", max);
(15) }
```

## Lignes 10 à 13

L'instruction `if`, qui commence à la **ligne 10** et se termine à la **ligne 13**, compare les deux valeurs saisies (`x >= y`).

Si (`if`) la valeur de `x` est supérieure à (`>=`) celle de `y`, alors (**ligne 11**) on affecte à `max` la valeur de `x` (`max = x`), sinon (`else`) – c'est-à-dire que la valeur de `x` est (strictement) plus petite que celle de `y` – on affecte à `max` la valeur de `y` (`max = y`).

## Exemple d'exécution du programme

L'exécution du programme se déroule de la façon suivante :

- 1 La fonction `main` est appelée.
- 2 Le message `x ?` est affiché (ligne 6).
- 3 Un nombre entier est saisi par l'utilisateur (supposons que ce soit 5) et affecté à la variable `x` (ligne 7).
- 4 Le message `y ?` est affiché (ligne 8).
- 5 Un nombre entier est saisi par l'utilisateur (supposons que ce soit 19) et affecté à la variable `y` (ligne 9).
- 6 La condition `x >= y` est testée (ligne 10) : elle est fausse car 5 n'est pas supérieur à 19. La clause `else` (lignes 12 et 13) est donc exécutée de sorte que l'on affecte la valeur 19 de `y` à `max`.
- 7 Le message `max = 19` est affiché à l'écran (ligne 14).

# Réalisation d'un programme

La réalisation d'un programme C se déroule en trois étapes :

- 1 Saisir le texte du programme, appelé **code source**, sous un éditeur de texte et le sauvegarder dans un fichier (**fichier source**).
- 2 Soumettre le fichier source à un **compilateur** C qui traduit le programme en code machine et produit un **fichier exécutable**. En salle TP cela peut être réalisé par la commande **gcc** qui produit un fichier exécutable **a.out**.
- 3 Pour **exécuter** le programme il suffit de taper la commande **./a.out**.

## Rédaction d'un programme

La disposition du texte d'un programme est libre. Des espaces, des tabulations et des retours à la ligne peuvent être placés à tout endroit où cela ne coupe pas un identificateur (un mot-clef, un nom de variables, ...).

Il est impératif, pour la lisibilité et la compréhension d'un programme, de l'**indenter** et de le **commenter**.

Un commentaire commence par `/*` et se termine par `*/`. Les commentaires peuvent s'étendre sur plusieurs lignes mais ne peuvent pas être imbriqués.