



R. Malgouyres, R. Zrour et F. Feschet  
Initiation à l'algorithmique et à la  
programmation en C,  
Cours avec 129 exercices corrigés, 3e édition,  
DUNOD, Collection Sciences Sup, 2014,  
Nouvelle présentation 2015.

## Algorithmique et programmation en C

# TP n° 1 Types et entrées-sorties

### Objectifs :

Le but de ce TP est de prendre contact avec l'environnement *linux*, un éditeur de texte, le compilateur *gcc*, et de réaliser un premier programme. On mettra ensuite en jeu le types de base du C et les entrées sorties de `stdio.h`.

## 1 Créer et compiler un programme

### 1.1 Se “loguer” :

Tapez votre login (votre nom d'utilisateur), puis à l'invite du système, votre mot de passe. Après quelques instants, le bureau apparaît avec la barre des tâches en bas de l'écran.

### 1.2 La console

Pour faire apparaître une console, cliquez sur l'icône représentant un écran d'ordinateur et un shell. Une fenêtre apparaît : c'est la console qui permet de commander le système linux.

Dans la console, tapez `ls` (abréviation de *list*). Le contenu de votre répertoire d'accueil apparaît.

```
$ ls
```

**Rappel :** Un fichier représente des données sur le disque. Chaque fichier porte un nom. Les fichiers sont regroupés dans des répertoire. Chaque répertoire peut contenir des fichiers, ou d'autres répertoires qui contiennent eux aussi des fichiers,...

Pour afficher le nom du répertoire courant, tapez `pwd`. (abréviation de *print working directory*)

### 1.3 Créer un répertoire

Pour créer un répertoire nommé “algorithmique” tapez

```
$ mkdir algorithmique
```

(abréviation de *make directory*.)

Pour aller dans le répertoire `algorithmique` tapez

```
$ cd algorithmique
```

(abréviation de *change directory*)

Vérifiez par `pwd`.

Créez ensuite un répertoire “`tp1`” et allez dedans en utilisant `cd`.

## 1.4 Lancer un éditeur de texte

Pour écrire un programme, il faut taper le programme dans un éditeur de texte (*kate*, *xemacs*, *nedit*, *vi*, etc... à votre convenance). Par exemple, pour lancer l'éditeur *xemacs* sur le fichier “`exercice1.c`”, on tape :

```
$ xemacs exercice1.c &
```

Dans l'éditeur , tapez un programme *C* qui affiche le message “bonjour” (voir chapitre ??). Pour sauvegarder, tapez (sous *xemacs*) `Ctrl-x Ctrl-s`. Le message “`wrote exercice1.c`” doit s'afficher en bas de la fenêtre. Dans la console, on peut vérifier en tapant `ls` que le fichier `.c` a bien été créé.



Ne pas oublier l'extension `.c` pour les fichiers source. Cela pourrait provoquer la perte des données lors de la compilation

## 1.5 Compiler et exécuter le programme

Avant de compiler un programme pour le tester, il faut toujours sauvegarder les dernières modifications. Pour pouvoir exécuter et tester un programme, il faut le *compiler*, ce qui génère un fichier exécutable à partir de votre code source. Pour cela, cliquez dans la console et tapez :

```
$ gcc exercice1.c -o exercice1
```

Deux cas peuvent se produire :

1. ou bien il y a un message d'erreur, qu'il faut essayer d'interpréter, qui indique une erreur dans votre code source (en précisant le numéro de la ligne de code où se trouve l'erreur). Corrigez l'erreur dans le programme *C* et recommencez... ;
2. ou bien il n'y a pas de message d'erreur et le compilateur a dû générer un fichier exécutable `exercice1` (vérifiez le par `ls`).

Vous pouvez exécuter le programme en tapant le nom de l'exécutable :

```
$ ./exercice1
```

## 2 Exercices de programmation

On donnera un nom de fichier différent à chaque programme pour chaque exercice.

**Exercice 1** Créer un programme qui calcule le prix d'une marchandise en appliquant un taux de réduction entré au clavier. Le prix hors réduction est aussi entré au clavier.

**Exercice 2** Dans une grande entreprise  $X$ , le salaire moyen des femmes est de 800 euros, et le salaire moyen des hommes est de 1100 euros. Écrire un programme qui calcule le pourcentage de rémunération en moins des femmes par rapport aux hommes dans l'entreprise  $X$ .

**Exercice 3** Évelyne a  $x$  ans. Tous les 7 ans, elle fait un pèlerinage sur son lieu de naissance : *vladivostok*. Combien a-elle fait de pèlerinages en tout ? Dans combien de temps doit-elle faire son prochain pèlerinage ? Écrire un programme qui résoud le problème pour un nombre  $x$  lu au clavier.

**Exercice 4** Écrire un programme qui prend en entrée une date sous la forme suivante :

1. l'année ;
2. numéro du jour dans l'année (exemple : le 5 février à pour numéro 36) ;

Le programme doit afficher un code du jour de la semaine : 0 pour lundi, 1 pour mardi, etc..., 6 pour dimanche.