

C/C++: Les fonctions.



Vincent Gaudillière

Polytech Nancy

2017 / 2018

Retours sur le TD n°1.

- Déclaration des variables au début du main (donc dans le main!),
- Quand on déclare une variable, elle peut prendre potentiellement n'importe quelle valeur. Donc ne pas oublier de l'initialiser si nécessaire (avant de l'incrémenter par exemple),
- Faire attention aux types des variables (cf "division par 0"),
- Un seul fichier source (+ commentaires),
- Respecter les indentations (lisibilité).

Environnement de travail.

- Utiliser Eclipse CDT (raccourci "Eclipse cpp"), et non pas Eclipse tout court (raccourci "Eclipse" ou "Eclipse Java"),
- Workspace au niveau du Z: (fait par défaut),
- S'assurer qu'il y a assez de place dans votre Z:,
- Bien utiliser MinGW GCC pour la compilation.

Sinon: <http://www.codeblocks.org/downloads/26>



Windows XP / Vista / 7 / 8.x / 10:

File

codeblocks-16.01-setup.exe

codeblocks-16.01-setup-nonadmin.exe

codeblocks-16.01-nosetup.zip

codeblocks-16.01mingw-setup.exe

codeblocks-16.01mingw-nosetup.zip

codeblocks-16.01mingw fortran-setup.exe

Les fonctions en C.

Une fonction, c'est:

- Un nom,
- Aucune, une, ou plusieurs variables en entrée (possiblement de types différents),
- Au plus une seule variable en sortie (mot clé `return`),
- Un bloc d'instructions.

```
type nomFonction(parametres)
{
    // instructions;
}
```

Exemples de fonctions.

```
float demi(float x)
{
    return x/2;
}
```

```
int addition(int a, int b)
{
    return a+b;
}
```

```
void erreur()
{
    printf("Erreur!\n");
}
```

Déclaration et appel d'une fonction.

```
int triple(int n) // (1) déclaration de la fonction
{
    return 3*n;
}

int main()
{
    int a,b;
    scanf("%d",a); // a==1 (par exemple)
    b = triple(a); // (2) appel de la fonction
    printf("%d\n.",b); // b==3
    return 0;
}
```

Fonctions et variables.

- La «durée de vie» d'une variable est limitée au bloc d'instructions dans lequel elle est déclarée.
- Une variable déclarée dans une fonction (`main` ou autre) n'est pas accessible hors de cette fonction.
- Mais on peut transférer des valeurs entre une fonction et l'«extérieur» (paramètres d'entrée et de sortie de la fonction).

Récurtivité.

- Une fonction peut s'appeler elle-même: on parle alors de **fonction réursive**.
- Toute fonction réursive doit posséder un cas limite (cas trivial) qui arrête la réursive.
- *Algorithmique*: utile lorsque la solution du problème général demande la résolution de plusieurs sous-problèmes particuliers, qui sont semblables au premier problème.

Récurtivité.

```
int factorielle(int n)
{
    if(n==0) // cas trivial
    {
        return 1;
    }
    else
    {
        return n*factorielle(n-1);
    }
}
```

Retour sur la déclaration d'une fonction.

```
type maFonction(parametres); // déclaration
{
    // instructions de maFonction
}

int main()
{
    // instructions du main
}
```

Prototype d'une fonction.

```
type maFonction(parametres); // prototype

int main()
{
    // instructions du main
}

type maFonction(parametres) // fonction
{
    // instructions de maFonction
}
```

Prototype d'une fonction.

```
int triple(int n); // prototype

int main()
{
    int a,b;
    scanf("%d",a); // a==1 (par exemple)
    b = triple(a); // appel de la fonction
    printf("%d\n.",b); // b==3
    return 0;
}

int triple(int n){ // fonction
    return 3*n;
}
```

Structure d'un projet.

- un fichier `mesFonctions.h` (*header*), contenant le prototype des fonctions,
- un fichier `mesFonctions.c`, contenant le code des fonctions,
- un fichier `main.c`, contenant le main.

Dans `main.c` et `mesFonctions.c`, il faut ajouter:

```
#include <stdio.h>
#include <stdlib.h>

#include "mesFonctions.h" // à ne pas oublier!

...

```

Le préprocesseur.

- mots-clés commençant par #: **directives de préprocesseur**,
- directives traitées juste **avant la compilation**,
- #include unFichier: ligne **remplacée** par le contenu du fichier.

```
#include <stdio.h> // directive de préprocesseur n°1
#include <stdlib.h> // dir. de préprocesseur n°2

#include "mesFonctions.h" // dir. de préprocesseur n°3

int main(){ // ...
```