



Définition et utilité



Programmation modulaire en C



- Découper le programme en plusieurs parties : sous-programmes ou modules.
- Chaque module peut, si nécessaire, être décomposé à son tour en modules plus élémentaires ;
- Ce processus de décomposition pouvant être répété autant de fois que nécessaire : c'est le principe de la **programmation structurée**.
- Regrouper dans le programme principal les instructions décrivant les enchaînements des modules.
- La seule sorte de module existant en langage C est **la fonction**.



Définition d'une fonction

Une fonction est un bloc d'instructions structurées autour d'un nom générique appelé "le nom de la fonction".

Ce bloc d'instructions peut éventuellement avoir besoin de valeurs pour pouvoir travailler sur des données qu'on lui transmet : on dit que la fonction peut avoir besoin de **paramètres**.



Intérêts des fonctions



- Améliorer la lisibilité du programme.
- Eviter des séquences d'instructions répétitives.
- Faciliter la maintenance.
- Faciliter la conception.
- Partager des outils communs qu'il suffit d'avoir écrits et mis au point une seule fois → Faciliter la réutilisabilité.



Intérêts des fonctions



- Possibilité de découper le programme source en plusieurs parties stockées dans des fichiers sources séparés. Chaque fichier source, appelé par abus de langage "module", regroupe une ou plusieurs fonctions et peut être compilé séparément
- Faciliter le travail en équipe + Développement de grosses applications.



Caractéristiques des fonctions



- Dans un programme, on distingue :
 - des fonctions **externes** intégrées dans des bibliothèques externes : `printf()`, `scanf()`, ...
 - des fonctions **internes** développées dans le programme : ce sont des fonctions personnalisées qui apportent une structuration au programme et qui **peuvent être appelées dans plusieurs fonctions.**
- Une fonction peut fournir un unique résultat (sa valeur de retour), comme ne pas en fournir du tout.



Caractéristiques des fonctions



- Une fonction peut retourner un résultat scalaire ou une structure
- La valeur d'une fonction peut dans certains cas ne pas être utilisée (comme les valeurs de retour de `printf()` et `scanf()`)
- Une fonction peut modifier les valeurs de certains de ses arguments (voir chapitre Pointeurs)

Utilisation des fonctions

1 Définition de la fonction

Fonction appelante

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

```
void NomFonction()
```

```
{
```

```
.....
```

```
}
```

```
void main()
```

```
{
```

```
int a, b, valmax;
```

```
printf(" Tapez deux entiers: ");
```

```
scanf("%d %d", &a, &b);
```

Appel de la fonction

2 Appel de la fonction par une autre fonction, ici par la fonction principale main().

```
NomFonction ();
```

```
.....
```

```
}
```