

## Fonctions

En C, une fonction peut être écrite de différentes manières :

K&R	ANSI
<pre>int max(a, b) int a, b; {     return (a&gt;b) ? a : b; }</pre> <p>❓ Si des paramètres sont omis dans la liste, ils sont considérés comme de type <i>int</i>.</p>	<pre>int max(int a, int b) {     return (a&gt;b) ? a : b; }</pre>

On préférera la forme ANSI puisque la forme K&R ne vérifie ni le type des paramètres ni leur nombre, et ne soulève donc ni warning ni erreurs.

## Retour

Il est intéressant de noter qu'une fonction peut retourner une structure et une union mais pas un tableau (possible à l'aide de pointeur), mais elle peut en modifier un (il est alors recommandé de donner comme valeur de retour *void*, soit rien).

## Paramètres

Les paramètres (ou arguments) d'une fonction sont moins restrictifs. On peut par exemple transmettre un tableau à une fonction.

Il est possible de passer un nombre variable d'arguments à une fonction :

```
#include <stdarg.h>
```

<pre>int evaluer(char operateur, int operande, ...){     va_list ap;     va_start(ap, operande);     int res=operande;     int arg;     while((arg=va_arg(ap, int))!=-1){         switch(operateur){             case '+':                 res += arg;                 break;             case '-':                 res -= arg;                 break;             case '*':                 res *= arg;                 break;             case '/':                 res /= arg;                 break;         }     }     return res; }</pre>	<p>Indique qu'il aura <i>n</i> arguments.</p> <p>Déclare une variable <i>va_list</i> (nécessite la bibliothèque <i>stdarg</i>).</p> <p>Démarre la liste avec <i>operande</i> (le 3<sup>e</sup> argument sera le premier à être retourné par <i>va_arg</i>).</p> <p>Retourne l'argument suivant (du type <i>int</i>) Ici il a été décidé que l'argument finale serait égale à -1 (ce n'est en aucun cas une généralité).</p>
--	---

## Déclarations

Pour utiliser une fonction il faut qu'on ait au minimum déclaré son prototype (~signature) auparavant dans le code (ou par le biais d'un fichier header la référençant, *.h*) :

K&R	ANSI
<pre>double cos(); //ne nécessite pas de préciser les paramètres</pre>	<pre>double cos(double x); //nécessite l'en-tête complète de la fonction</pre>