

Langage C : Guide de style élémentaire

Les types utilisateurs : Les noms des types définis par l'utilisateur se terminent par "_t". Par exemple "complex_t" ou "boolean_t".

Les constantes : Les constantes définies par l'utilisateur sont en majuscule. Par exemple : **#define PI 3.14**. On doit toujours essayer de nommer les constantes.

Les fonctions : Chaque fonction est précédée d'un bloc de commentaires précisant le nom de la fonction, son rôle, les paramètres en entrée et en sortie ainsi que sa valeur de retour. Les fonctions C qui ne retournent rien sont déclarées comme ayant le type **void**. Par exemple :

```
/* ----- */
/* strcmp          Compare deux chaînes de caractères      */
/*                                                         */
/* En entrée: s1, s2 Deux chaînes de caractères           */
/*                                                         */
/* En sortie: La valeur entière retournée est 0 si les chaînes sont */
/*           égales; négative si s1 < s2 et positive si s1 > s2. */
/* ----- */
```

Déclarations : Les déclarations sont alignées suivant leurs types. Les pointeurs sont décalés d'un caractère. Par exemple :

```
int          i, j;
struct adresse Adresse_Benny;
char         * ptr;
float        x = 10.45,
            y = 20;
```

Commentaires : Les commentaires sont alignés. Les sources sont aérées autant que possible.

Indentation : Les sources sont indentées de manière cohérente et rigoureuse. Des exemples peuvent être pris sur les sources issus de 'cb' (formateur de source C : **c** beautiful). Exemple d'indentation avec 3 caractères.

```
if (condition)      while(condition)      for(condition)      do
{
  Bloc alors        {      Bloc Tant que      {      Bloc pour      {      Bloc répéter
}
else
{
  Bloc sinon
}
}
```

Fichiers : Un fichier commence par un bloc de commentaires donnant le nom du fichier et son rôle, que ce soit un entête (fichier .h : déclaration des types, constantes et prototypes) ou un module (fichier .c).

ATTENTION: Les fichiers d'entête (.h) ne doivent contenir que : (1) des déclarations de types, (2) des déclarations de constantes, (3) des déclarations de macro-instructions, (4) des déclarations de prototypes de fonctions, (5) des inclusions d'autres fichiers d'entête (pas de fichiers .c !), des déclarations d'accès à des variables externes ou (6) des directives de compilation conditionnelles.

En aucun cas ils ne doivent contenir de code C exécutable ou des déclarations de variables.