**Les en-têtes de la bibliothèque C ISO**

**<assert.h>**

Contient la macro assert, utilisée pour aider à détecter des incohérences de données et d'autres types de bogues dans les versions de débogage d'un programme.

**<complex.h>**

Pour manipuler les nombres complexes (introduit par C99).

**<ctype.h>**

Fonctions utilisées pour classifier rapidement les caractères, ou pour convertir entre majuscules et minuscules de manière indépendante du système de codage des caractères (character set) utilisé ([ASCII](https://fr.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange), [ISO/CEI 8859-1](https://fr.wikipedia.org/wiki/ISO/CEI_8859-1), [EBCDIC](https://fr.wikipedia.org/wiki/Extended_Binary_Coded_Decimal_Interchange_Code), etc.).

**<errno.h>**

Ensemble (ou le plus souvent sous-ensemble) des codes d'erreurs renvoyés par les fonctions de la [bibliothèque](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) standard au travers de la variable errno.

**<fenv.h>**

Pour contrôler l'environnement en virgule flottante (floating-point) (introduit par C99).

**<float.h>**

Contient des constantes qui spécifient les propriétés des nombres en virgule flottante qui dépendent de l'implémentation, telles que la différence minimale entre deux nombres en virgule flottante différents (xxx\_EPSILON), le nombre maximum de chiffres de précision (xxx\_DIG) et l'intervalle des nombres pouvant être représentés (xxx\_MIN, xxx\_MAX).

**<inttypes.h>**

Pour des conversions précises entre types entiers (introduit par C99).

**<iso646.h>**

Pour programmer avec le jeu de caractères ISO 646 (introduit par Amd.1).

**<limits.h>**

Contient des constantes qui spécifient les propriétés des types entiers qui dépendent de l'implémentation, comme les intervalles des nombres pouvant être représentés (xxx\_MIN, xxx\_MAX).

**<locale.h>**

Pour s'adapter aux différentes conventions culturelles.

**<math.h>**

Pour calculer des fonctions mathématiques courantes. C99 a ajouté de nombreuses fonctions mathématiques, en particulier pour converger avec la norme CEI 559 dite aussi IEEE 754.

**<setjmp.h>**

Pour exécuter des instructions goto non locales (sortes d'exceptions).

**<signal.h>**

Pour contrôler les *signaux* (conditions exceptionnelles demandant un traitement immédiat, par exemple signal de l'utilisateur).

**<stdarg.h>**

Pour créer des fonctions avec un nombre variable d'arguments.

**<stdbool.h>**

Pour avoir une sorte de type booléen (introduit par C99).

**<stddef.h>**

Définit plusieurs types et macros utiles, comme NULL.

**<stdint.h>**

Définit divers types d'entiers, c'est un sous-ensemble de inttypes.h (introduit par C99).

**<stdio.h>**

Fournit les capacités centrales d'entrée/sortie du langage C, comme la fonction printf.

**<stdlib.h>**

Pour exécuter diverses opérations dont la conversion, la génération de nombres pseudo-aléatoires, l'allocation de mémoire, le contrôle de processus, la gestion de l'environnement et des signaux, la recherche et le tri.

**<string.h>**

Pour manipuler les chaînes de caractères.

**<tgmath.h>**

Pour des opérations mathématiques sur des types génériques (introduit par C99).

**<threads.h>**

Implémente la gestion de threads comme un standard de la libc (introduit par C11)

[**<time.h>**](https://fr.wikipedia.org/wiki/Time.h)

Pour convertir entre différents formats de date et d'heure.

**<wchar.h>**

Pour manipuler les caractères larges (wide char), nécessaire pour supporter un grand nombre de langues et singulièrement Unicode (introduit par Amd.1).

**<wctype.h>**

Pour classifier les caractères larges (introduit par Amd.1).