

Aide C :

1. FONCTIONS D'ENTRÉE/SORTIE

- **printf** : affiche une chaîne de caractères à l'écran
- **scanf** : lit une valeur depuis l'entrée standard (clavier)
- **gets** : lit une chaîne de caractères depuis l'entrée standard (clavier)
- **putchar** : affiche un caractère à l'écran
- **puts** : affiche une chaîne de caractères suivie d'un saut de ligne

2. FONCTIONS DE MANIPULATION DE CHÂÎNES DE CARACTÈRES

- **strlen** : renvoie la longueur d'une chaîne de caractères
- **strcpy** : copie une chaîne de caractères dans une autre
- **strcat** : concatène deux chaînes de caractères
- **strcmp** : compare deux chaînes de caractères

3. FONCTIONS MATHÉMATIQUES

- **abs** : renvoie la valeur absolue d'un nombre
- **ceil** : renvoie le plus petit entier supérieur ou égal à un nombre décimal
- **floor** : renvoie le plus grand entier inférieur ou égal à un nombre décimal
- **pow** : calcule la puissance d'un nombre
- **sqrt** : calcule la racine carrée d'un nombre

4. FONCTIONS DE GESTION DE LA MÉMOIRE

- **malloc** : alloue de la mémoire dynamiquement
- **calloc** : alloue de la mémoire dynamiquement et initialise les valeurs à 0
- **realloc** : modifie la taille de la mémoire allouée dynamiquement
- **free** : libère la mémoire allouée dynamiquement

5. COMMANDES DE CONTRÔLE DE FLUX

- **if/else** : permet d'exécuter du code conditionnellement
- **switch/case** : permet d'exécuter du code selon la valeur d'une expression
- **for** : permet d'exécuter du code de manière répétée selon une condition
- **while/do-while** : permet d'exécuter du code de manière répétée tant qu'une condition est vraie
- **break** : permet de sortir d'une boucle
- **continue** : permet de passer à la prochaine itération d'une boucle

6. AUTRES FONCTIONS ET COMMANDES

- **typedef** : permet de créer un alias pour un type de données existant
- **struct** : permet de créer une structure de données personnalisée
- **enum** : permet de créer un type de données énuméré
- **const** : permet de déclarer une variable constante
- **sizeof** : renvoie la taille d'un type de données en octets
- **return** : permet de retourner une valeur à partir d'une fonction

`#include <..... .h>` = Bibliothèques utiles pour le code (code tout prêt)

`int main()` = Fonction principale d'un programme (on peut en créer d'autres)
{ }

`printf("Hello world!\n");` = instructions présente dans une fonction (ici, `printf("");`)

`return 0;` = Fin du programme, en envoi un 0 pour dire que tout s'est bien déroulé (facultatif)

```
#include <stdio.h>
#include <stdlib.h> } Directives de préprocesseur

int main()
{
printf("Hello World !\n");
return 0;
} } Instructions } Fonction
```

`printf("");` = permet d'afficher un message à l'écran (affiche ce qui est entre les guillemets)
(ex : `printf(" Quel âge avez-vous ? ");`).

- ajoutez « \n » entre les guillemets pour aller à la ligne.

- ajoutez « %... » entre les guillemets pour afficher la valeur d'une variable, il est possible d'afficher plusieurs valeurs de plusieurs variables.
(ex : `printf("Il vous reste %d vies", Variable1);` la lettre correspond au type de la variable utilisée)

Format	Type attendu
"%d"	int
"%u"	unsigned int
"%ld"	long
"%f"	float
"%lf"	double

`scanf("%...", &...);` = Permet de saisir et d'affecter manuellement une valeur à une variable
(ex: `scanf("%d", &age);`).

`//`, `/* */` = Permet d'ajouter un commentaire

`int, float, double, char, long...` = Noms de plusieurs types variables associées à un nom choisi
(ex : `int Variable1;`).

- On peut initialiser plusieurs nom pour un même type de variable
(ex : `float Variable1, Variable2;`)
- utilisez « unsigned ... » avant le type pour stocker que des nombres positifs
(ex: `unsigned int Variable1;`)
- utilisez « const ... » avant le type d'une variable pour lui affecter une valeur fixe qui ne changera jamais. (ex : `const double Variable1 = 5;`)

unsigned char	0 à 255
unsigned int	0 à 65 535
unsigned long	0 à 4 294 967 295

Nom du type	Minimum	Maximum
signed char	-127	127
int	-32 767	32 767
long	-2 147 483 647	2 147 483 647
float	-1 x1037	1 x1037
double	-1 x1037	1 x1037