



STRUCTURE DE BASE D'UN PROGRAMME C

Anatomie d'un programme en langage C

```
/*
 * Programme: CalculSalaire
 * Ce programme saisit le taux horaire d'un employé et le nombre d'heures travaillées,
 * calcule le salaire à verser et l'affiche à l'écran.
 *
 * Auteur: Vincent Lacasse
 * Date: 2016-08-31
 * Version 1.0
 */
```

← Commentaires

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

← Commandes de préprocesseur

int main(void) ← Point d'entrée au programme

```
{
    double salaire_horaire; /* Salaire horaire de l'employé. */
    int nb_heures;          /* Nombre d'heures travaillées. */
    double total;          /* Salaire à verser à l'employé. */

    /* On saisit le salaire horaire. */
    printf("Veuillez entrer le salaire horaire : ");
    scanf("%lf", &salaire_horaire);

    /* On saisit le nombre d'heures travaillées. */
    printf("Veuillez entrer le nombre d'heures de travail : ");
    scanf("%i", &nb_heures);

    /* On calcule le salaire à verser. */
    total = nb_heures * salaire_horaire;

    /* On affiche le salaire à verser à l'employé. */
    printf("Le salaire à verser est : %.2f$\n", total);
    return EXIT_SUCCESS;
}
```

} Déclarations de variables
(et commentaires explicatifs)

} Instructions
(exécutées de haut en bas)

COMMANDES DE PRÉPROCESSEUR

Syntaxe:

`#include <fichier_de_la_librairie_standard>`

OU

`#include "fichier_utilisateur"`

Syntaxe (de base):

`#define IDENTIFICATEUR valeur_de_replacement`

FORMATS DE COMMENTAIRES

- Il existe deux formats: en-ligne et en bloc.
- **Commentaires en-ligne:** Doit obligatoirement tenir sur une ligne. Peut se trouver suite à une instruction

```
//Calcul du total
```

```
z = x*x % y; //Selon la formule de Yves Bergeron
```

- **Commentaire en bloc:** Peuvent s'étendre sur plusieurs lignes, sont délimités par /* et */

```
/* Programme: Calcul du salaire  
Auteur: Vincent Lacasse */
```

DÉCLARATION D'UNE VARIABLE

- **Avant d'utiliser** une variable nous devons la déclarer
- C'est par ce procédé que le système réserve l'espace mémoire nécessaire pour stocker l'information de la variable
- En C, la déclaration doit se faire au début d'un bloc de fonction

Syntaxe:

```
type identifiant[=valeur_initiale];
```

Exemples:

```
int credits=4; /* Nombre de crédits du cours, 4 par défaut */  
char note; /* La note finale obtenue */
```

IDENTIFIANT D'UNE VARIABLE

- L'identifiant doit respecter les règles suivantes:
 - Ne peut être composé que de **lettres alphabétiques non accentuées**, de **chiffres** ou du **caractère de soulignement** (_).
 - Doit obligatoirement commencer par une lettre alphabétique
 - Ne doit pas être un mot réservé du langage



Conventions:

- Un nom de variable commence par une lettre minuscule;
- Si le nom se compose de plusieurs mots, on sépare les mots par un _ (snail case), par exemple:
nombre_heures_travaillees
- Pour faciliter la compréhension, le nom de variable **doit** représenter son contenu.

TYPES ENTIERS (SOUS VISUAL STUDIO)

type	taille	signed	unsigned
bool	1 bit	0 → 1	<i>Non défini</i>
char	1 octet	-128 → 127	0 → 255
short	2 octets	-32768 → 32767	0 → 65535
int	4 octets	-2 147 483 648 → 2 147 483 647	0 → 4 294 967 296
long	4 octets	-2 147 483 648 → 2 147 483 647	0 → 4 294 967 296
long long	8 octets	-9.2 x 10 ¹⁸ → 9.2 x 10 ¹⁸	0 → 1.8 x 10 ¹⁹

Réf.: <https://msdn.microsoft.com/en-us/library/s3f49ktz.aspx>

TYPES RÉELS (SOUS VISUAL STUDIO)

type	taille	Valeurs
float	4 octets	$1.175494351 \times 10^{-38}$ → $3.402823466 \times 10^{+38}$
double	8 octets	$2.2250738585072014 \times 10^{-308}$ → $1.7976931348623158 \times 10^{+308}$
long double	8 octets	$2.2250738585072014 \times 10^{-308}$ → $1.7976931348623158 \times 10^{+308}$

Réf.: <https://msdn.microsoft.com/en-us/library/s3f49ktz.aspx>

LES ENTRÉES/SORTIES

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

Syntaxe:

```
printf("format", valeurs_positionnelles... );
```

Exemple:

```
printf("Montant:%lf $CAN = %lf $US", montant_cad, montant_us);
```



Syntaxe:

```
scanf("format", &variable );
```

Exemple:

```
scanf("%d", nb_etudiants );
```

FORMATS FRÉQUENTS

Types	Format
char	%c
short int	%hd ou %hi
unsigned short int	%hu
int	%d ou %i
unsigned int	%u
long int	%ld ou %li
unsigned long int	%lu
float	%f
double	%lf
long double	%lf

CARACTÈRES D'ÉCHAPPEMENT

Séquence d'échappement	Signification
<code>\a</code>	Produit une alarme sonore
<code>\b</code>	Appuie sur la touche retour en arrière
<code>\f</code>	Insère un saut de page
<code>\n</code>	Insère un saut de ligne
<code>\r</code>	Ramène le curseur d'écriture au début de la ligne courante.
<code>\t</code>	Insère une tabulation horizontale
<code>\v</code>	Insère une tabulation verticale
<code>\\</code>	Affiche le symbole <code>\</code>
<code>\'</code>	Affiche le symbole <code>'</code>
<code>\"</code>	Affiche le symbole <code>"</code>
<code>\?</code>	Affiche le symbole <code>?</code>
<code>\nnn</code>	Affiche le symbole ASCII correspondant au code octal <code>nnn</code>
<code>\xhh</code>	Affiche le symbole ASCII correspondant au code hexadécimal <code>hh</code>