

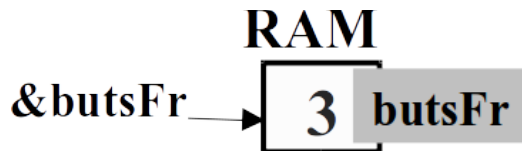


# Algorithmique & Programmation en C

## DECLARER VARIABLE

### Ressources :

[Diapos MEMORISER](#)



### Objectif :

- Connaître la notion de *Variable*.
- Comment déclarer une variable dans un programme ?

## 1. DEFINITION

Une **variable** est une **donnée** qui sert à manipuler des grandeurs de diverses natures (valeurs entières, réelles, caractères...) dans un programme. Les valeurs des variables sont **stockées dans la RAM** lorsque le programme est en cours d'exécution. La particularité d'une variable est que sa **valeur peut varier** en cours d'exécution du programme : elle peut être changée plusieurs fois par les instructions d'un programme.

Une variable, lors de l'exécution du programme, est décrite par :

- × un **nom** (étiquette associée à une zone mémoire RAM -ex. *butsFr* sur le schéma en haut-) ;
- × un **valeur** (contenu 3 dans la zone mémoire sur le schéma) ;
- × une **adresse dans la RAM** (**pointeur** *&butsFr* sur la zone RAM de la variable *butsFr*).

Pour caractériser une variable, il faut lui associer un **type de donnée**, qui indique quelle est sa nature :

- × sur combien de bits les valeurs de la variable sont-elles codées (par exemple, une **taille de sa zone mémoire** de 2 octets est associée à un type « petit entier naturel ») ?
  - × les valeurs de la variable commencent-elles à 0 (**type non signé** : entier naturel) ou à une valeur négative (**type signé** : entier relatif) ?
  - × quel est l'ensemble des valeurs possibles pour ma variable ou quel est le **domaine de capacité** (par exemple, une variable de type « petit entier naturel » a des valeurs comprises dans  $[0, 65535]$ ) ?
- ATTENTION aux DEPASSEMENT de CAPACITE !!**
- × quels sont les opérateurs applicables à la variable (par exemple, + - / \* et % pour des entiers) ?

Les types simples du langage C :

TYPE	Signification	CAPACITE	TAILLE (en octets)
char / unsigned char	caractère	$[-128, 127] / [0, 255]$	1
unsigned short	Entier non signé court	$[0, 65\ 535]$	2
short	Entier signé court	$[-32\ 768, 32\ 767]$	2
unsigned long ou unsigned int	Entier non signé long	$[0, 4\ 294\ 967\ 295]$	4
long ou int	Entier signé long	$[-2147483648, 2147483647]$	4
float	Réel simple précision	$[+/- 3.4 \cdot 10^{38}, +/- 3.4 \cdot 10^{38}]$	4
double	Réel double précision	$[+/- 1.7 \cdot 10^{308}, +/- 1.7 \cdot 10^{308}]$	8

## 2. ALGORITHME (DÉCLARATION VARIABLE)

L'instruction de déclaration de variable permet d'indiquer les variables que le programme pourra manipuler. Elle s'écrit dans la *partie données* de la fonction qui utilise cette variable :

```

/*****
                                DECLARATION DE VARIABLE
*****/

ALGO DeclarerVariable    // PARTIE DONNEES
    VAR nomVariable : type // variables de type numérique ou caractère
        compt1 : ENTIER non signé court
        compt2 : ensc
        compt3, compt4 : unsigned short
DEBUT
FIN
  
```



### Règle de programmation : *nommage des variables*

- Noms explicites, pas trop longs ;
- 1<sup>o</sup> lettre en minuscule (variable locale) ;
- Préciser rôle variable avec un commentaire explicatif.

## 3. CODAGE EN LANGAGE C (DÉCLARATION VARIABLE)

La traduction en C consiste à indiquer d'abord le type de la variable, puis son nom suivi d'un ; :

```

/*****
                                DECLARATION DE VARIABLE
*****/

int main()
{
    // PARTIE DONNEES
    unsigned short    nomVariable1, nomVariable2; // TYPE numérique ou caractère
    // PARTIE INSTRUCTIONS    ...
}
  
```

L'exécution, par le CPU de l'instruction de déclaration de variable entraîne la réservation d'un espace mémoire vide (sans valeur), tel que :

- ✗ sa longueur (nombre d'octets) dépend du type choisi ;
- ✗ son nom est *nomVariable* et son adresse est *&nomVariable*.

