

# Programmation C

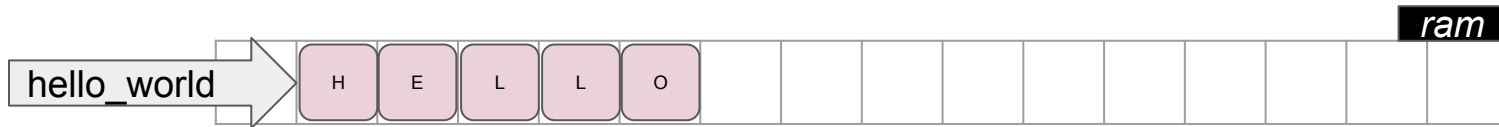
## 11 - Les strings

## les tableaux de char

### Exemple:

```
char hello_world[5] = { 'H', 'E', 'L', 'L', 'O' };
```

H	E	L	L	O
---	---	---	---	---

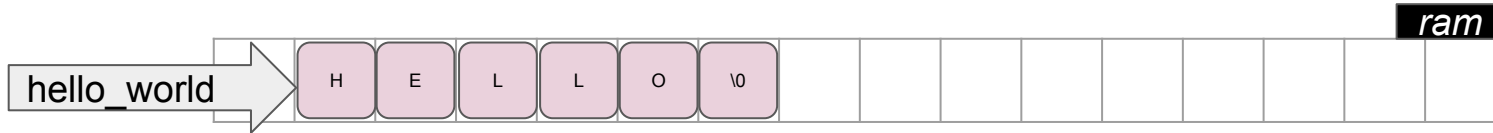


## Les strings

### Exemple:

```
char hello_world[6] = { 'H', 'E', 'L', 'L', 'O', '\0' };  
printf("%s", hello_world) -> HELLO
```

H	E	L	L	O	\0
---	---	---	---	---	----



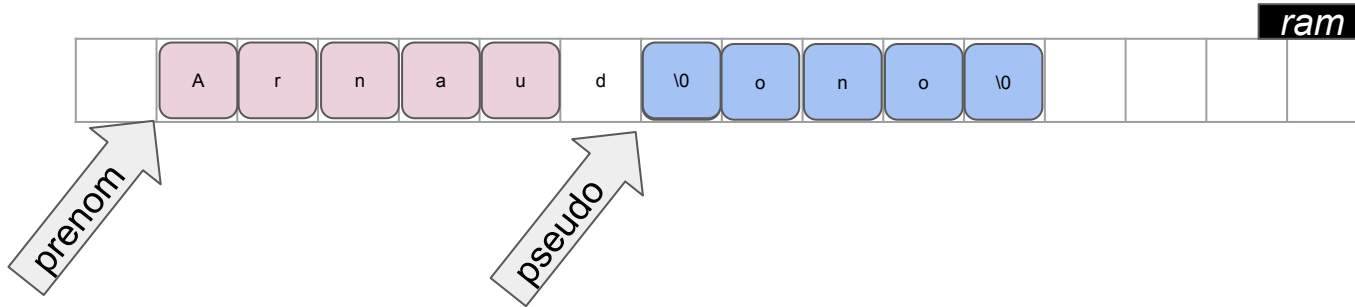
## Dépassement de mémoire

### Exemple:

```
char pseudo[5];  
pseudo = "Nono";
```

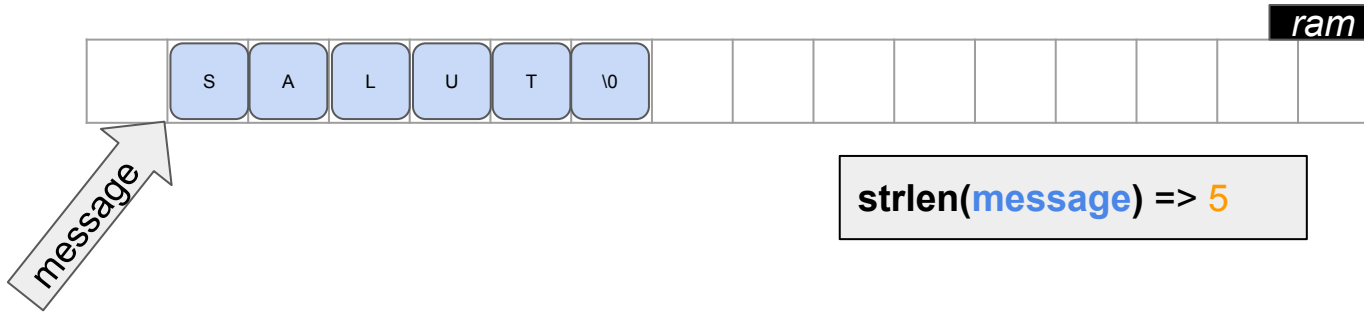
```
char prenom[5];  
prenom = "Arnaud";
```

```
printf("%s", prenom); => Arnaud  
printf("%s", pseudo); =>
```



## strlen (string.h)

taille strlen( chaîne )

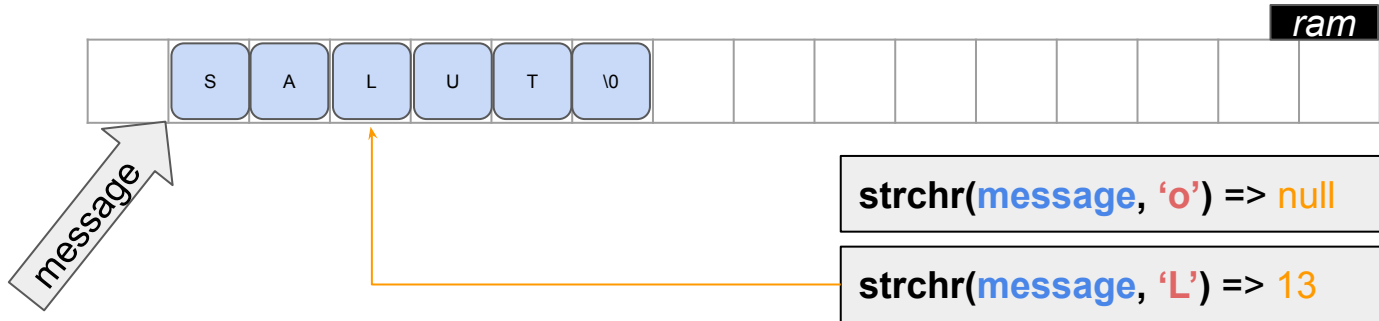


Retourne la taille de la string passé en paramètre en excluant le ' '

## strchr (string.h)

pointeur strchr( chaîne , char )

Adresse: 10 11 12 13 14 15 16 17 18 19



**Retourne la première sous chaîne qui commence par le caractère passé en argument.  
Si le Caractère n'est pas trouvé, elle retourne NULL.**

## strcmp (string.h)

```
int strcmp( chaîne 1, chaîne 2 )
```

```
strcmp("Tata", "Toto") => -1
```

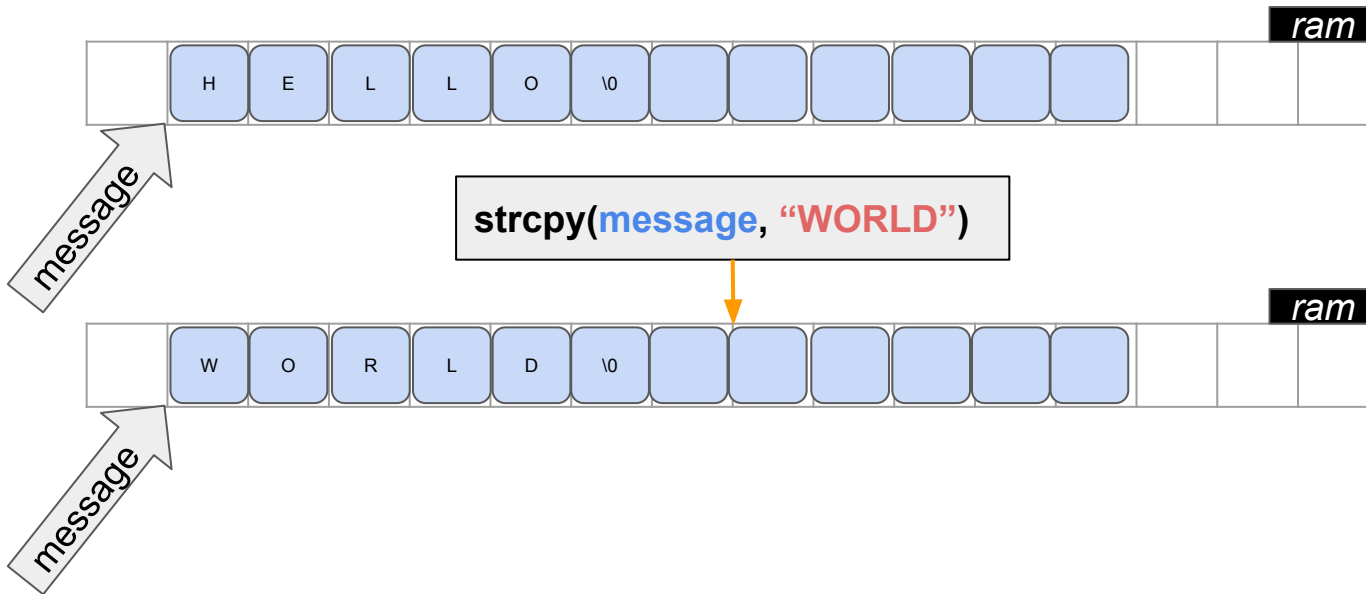
```
strcmp("Toto", "Toto") => 0
```

```
strcmp("Toto", "Tata") => 1
```

Compare deux chaîne, retourne 0 si elles sont identiques, sinon elle retourne -1 dans le cas ou la première chaîne est avant dans l'ordre alphabétique et elle retourne 1 dans l'autre cas.

## strcpy (string.h)

strcpy( chaîne 1 , chaîne 2 )

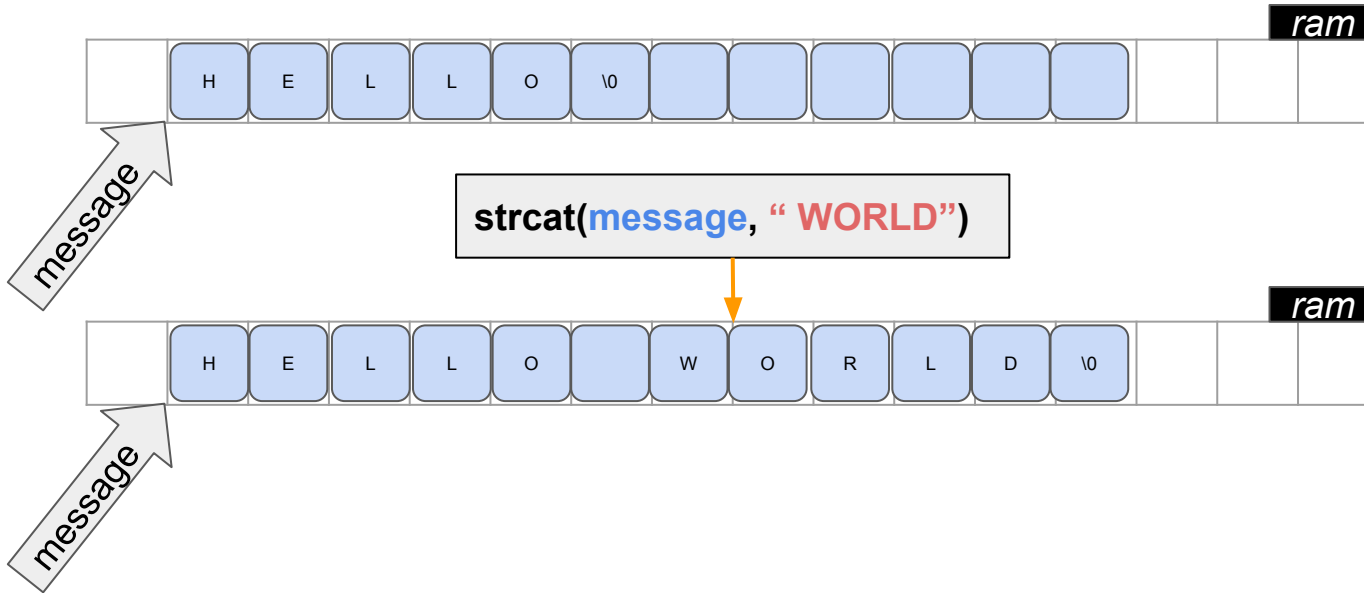


**Copie le contenu de la chaîne 2 dans la chaîne 1. Attention à ce que le tableau de chaîne 1 soit suffisamment grand pour contenir chaîne 2.**



## strcat (string.h)

strcat( chaîne 1 chaîne 2 )



Insère le contenu de la chaîne 2 devant la chaîne 1. Attention à ce que le tableau de chaîne 1 soit suffisamment grand.

## strtol(string.h)

`long` `strtol`( `chaîne` , `pt_fin` , `base` )

`strtol("123456", NULL, 10) => 123456`

`strtol("12€", NULL, 10) => 12`

**Convertit une chaîne en la valeur numérique entière qu'elle représente.**

**pt\_fin:** la fonction s'en sert pour renvoyer la position du premier caractère qu'elle a lu et qui n'était pas un nombre. On peut mettre NULL si on ne souhaite pas l'utiliser.

**base:** base avec laquelle le nombre est écrit dans la chaîne. le plus souvent base 10

## strtod(string.h)

double strtod( chaîne , pt\_fin )

strtod("12.3", NULL) => 12.3

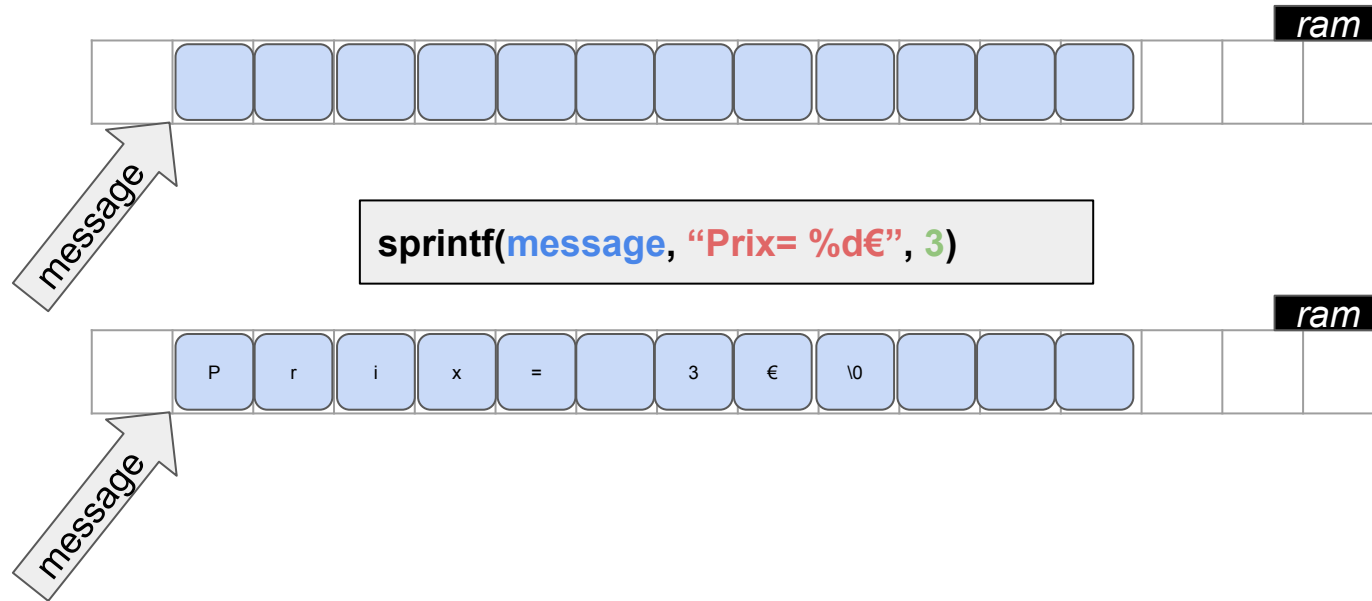
strtod("12.2€", NULL) => 12.2

**Convertit une chaîne en la valeur numérique flottante qu'elle représente.**

**pt\_fin: la fonction s'en sert pour renvoyer la position du premier caractère qu'elle a lu et qui n'était pas un nombre. On peut mettre NULL si on ne souhaite pas l'utiliser.**

## sprintf(stdio.h)

`sprintf( chaîne destination , chaîne template , params )`



Comme la fonction `printf`, `sprintf` permet de formater une chaîne de caractère mais au lieu de l'afficher dans la console, on l'écrit dans un tableau de char (string).

## lecture sécurisée avec fgets

retour fgets( chaîne , taille , stdin)

```
int TAILLE= 100;
char message[TAILLE];
printf("Ecrire un message de %d caractères maximum\n > ");

if( fgets(message, TAILLE, stdin) != NULL)
{
    printf("\n %s \n", message);
}
else
{
    printf("Une erreur est survenue \n");
}
```

*Exemple*

Ecrire un message de 100  
caractères maximum  
> Bonjour et bienvenue dans cette  
formation sur le C

Bonjour et bienvenue dans cette  
formation sur le C

fgets retourne NULL si il y a une erreur lors de la lecture