

La Programmation en langage C

Chapitre 7 : les chaînes de caractères

La cible : SMI S3

Pr S.ELFILALI

sanaa.elfilali@etu.univh2c.ma

les chaines de caractères

Que signifie une chaine de caractères ?

Une suite logique de caractères et qui vont constituer des mots ou des phrases

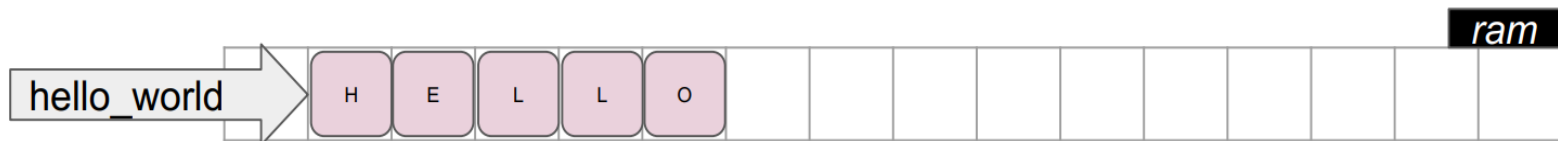
NB : En général en programmation il y a le type string qui permet de stocker des chaines de caractères mais en langage C , ce type de variable n'est pas disponible ! Mais on peut manipuler ou utiliser le string en C

Les tableaux de char

Exemple:

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

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



Les tableaux de char

Illustration sur CodeBlocks

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

int main()
{
    int i;
    char salut[]={'S','A','L','U','T'};
    for(i=0;i<5;i++)
    {
        printf("%c",salut[i]);
    }

    return 0;
}
```

```
SALUT
Process returned 0 (0x0)   execution time : 0.026 s
Press any key to continue.
```

Les strings

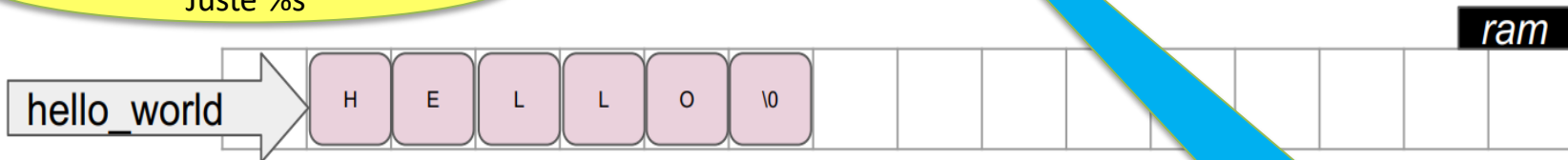
Exemple:

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

Pas besoin de boucle !!!
Juste %s

Un marqueur
Un caractère spécial

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



La fin de la chaîne
Pas la fin du tableau !

Les strings

Illustration sur CodeBlocks

```
int main()
{
    int i;
    char salut[]={ 'S','A','L','U','T','\0'};

    printf("%s",salut);

    return 0;
}
```

```
int main()
{
    int i;
    char salut[]={ 'S','A','L','\0','U','T','\0'};

    printf("%s",salut);

    return 0;
}
```

Si on met \0 ici

```
SALUT
Process returned 0 (0x0)   execution time : 0.026 s
Press any key to continue.
```

```
SAL
Process returned 0 (0x0)   execution time : 0.078 s
Press any key to continue.
```

Les strings

Illustration sur CodeBlocks

```
int main()
{
    int i;
    char salut[] = "SALUT";
    printf("%s",salut);

    return 0;
}
```

Initialisation avec une chaîne de caractère directement

```
SALUT
Process returned 0 (0x0)   execution time : 0.026 s
Press any key to continue.
```

```
int main()
{
    int i;
    char salut[]="SALUT";
    char prenom[100];
    scanf("%s", prenom);
    printf("%s, %s",salut, prenom);
    return 0;
}
```

```
Sanaa
SALUT, Sanaa
Process returned 0 (0x0)   execution time : 7.637 s
Press any key to continue.
```

Si vous mettez un prénom composé, il prend en considération que le premier mot!! (l'espace est une validation comme entrée

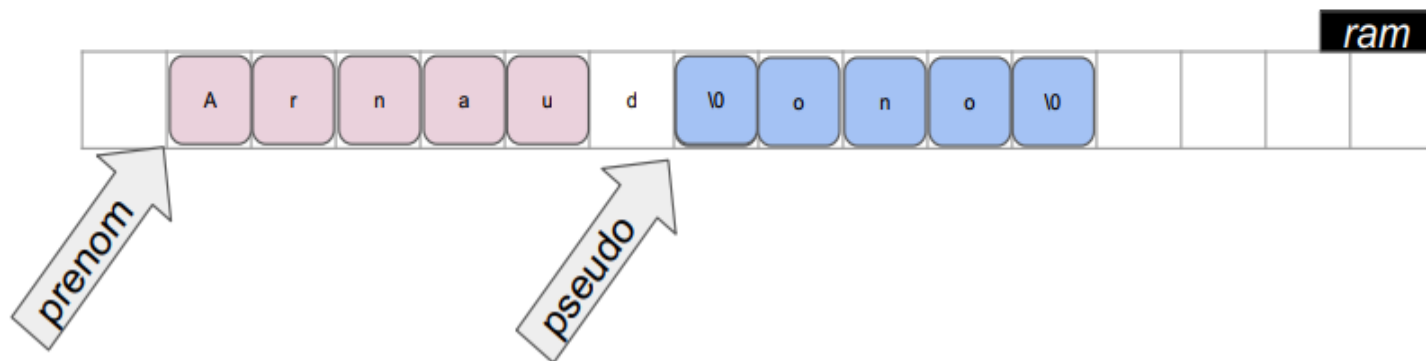
Dépassement de mémoire

Exemple:

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

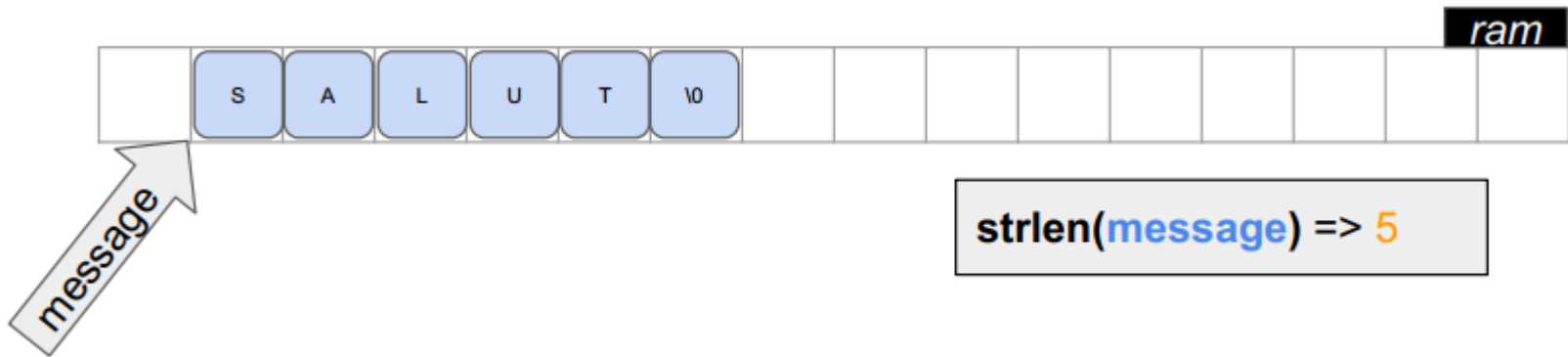
```
char prenom[5];  
prenom = "Arnaud";
```

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



Strlen(string.h)

taille strlen(chaîne)



[C documentation — DevDocs](https://devdocs.io/c/) : <https://devdocs.io/c/>

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

Strlen(string.h)

Illustration sur CodeBlocks

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

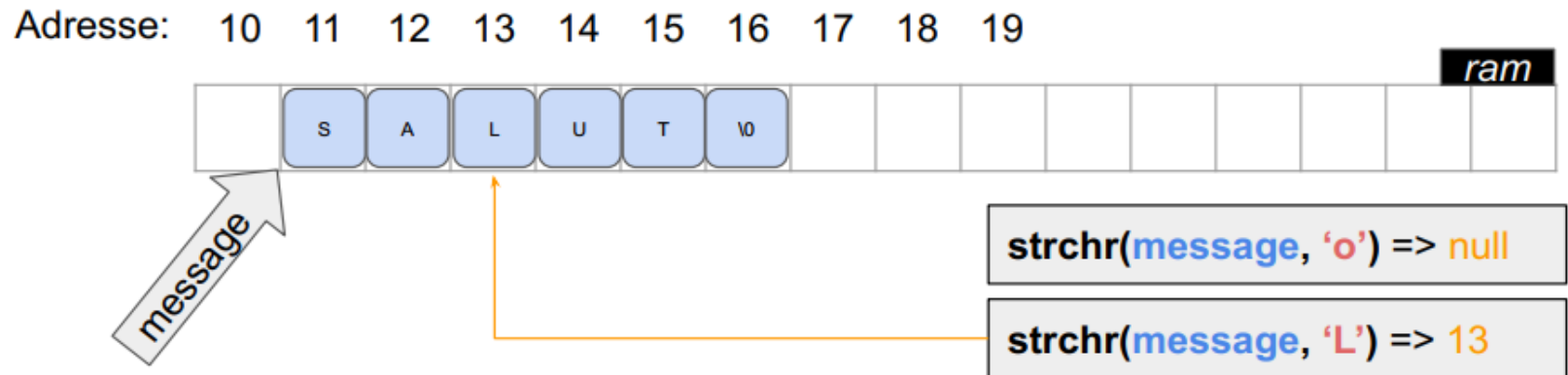
```
int main()
{
    char message[] = "hello world";
    printf("taille de message \" %s \" = %d \n",message, strlen(message));

    return 0;
}
```

taille de message " hello world " = 11

Strchr(string.h)

pointeur strchr(chaîne , char)



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

Strchr(string.h)

Illustration sur CodeBlocks

```
int main()
{
    char message[] = "hello world";
    char char_a_trouver = 'v';
    char* p_char = strchr(message ,char_a_trouver);

    printf("taille de message \" %s \" = %d \n",message, strlen(message));

    if (p_char ==NULL)
    {
        printf("la char \'%c\' n'est pas present dans la chaine \" %s \"",char_a_trouver,message);
    }
    else
    {
        printf("la char \'%c\'est present dans la chaine \" %s \" ",char_a_trouver,message)
    }
    return 0;
}
```

taille de message " hello world " = 11
la char 'v' n'est pas present dans la chaine " hello world "

Strchr(string.h)

Illustration sur CodeBlocks

```
int main()
{
    char message[] = "hello world";
    char char_a_trouver = 'e';
    char* p_char = strchr(message ,char_a_trouver);

    printf("taille de message \" %s \" = %d \n",message, strlen(message));

    if (p_char ==NULL)
    {
        printf("la char \'%c\' n'est pas present dans la chaine \" %s \"",char_a_trouver,message);
    }
    else
    {
        printf("la char \'%c\'est present dans la chaine \" %s \" ",char_a_trouver,message);
    }
    return 0;
}
```

taille de message " hello world " = 11
la char 'e'est present dans la chaine " hello world "

Strchr(string.h)

Illustration sur CodeBlocks

```
int main()
{
    char message[] = "hello world";
    char char_a_trouver = 'e';
    char* p_char = strchr(message + 3, char_a_trouver);

    printf("taille de message \" %s \" = %d \n", message, strlen(message));

    if (p_char == NULL)
    {
        printf("la char \" %c \" n'est pas present dans la chaine \" %s \"", char_a_trouver, message);
    }
    else
    {
        printf("la char \" %c \" est present dans la chaine \" %s \" ", char_a_trouver, message);
    }
    return 0;
}
```

taille de message " hello world " = 11
la char 'e' n'est pas present dans la chaine " hello world "

La recherche commence à partir du troisième caractère pour faire la recherche dans un sous ensemble de chaine

Strchr(string.h)

Illustration sur CodeBlocks

```
int main()
{
    char message[] = "hello world";
    char char_a_trouver = 'o';
    char* p_char = strchr(message + 3, char_a_trouver);

    printf("taille de message \" %s \" = %d \n", message, strlen(message));

    if (p_char == NULL)
    {
        printf("la char \'%c\' n'est pas present dans la chaine \" %s \" ", char_a_trouver, message);
    }
    else
    {
        printf("la char \'%c\' est present dans la chaine \" %s \" la fin du message apres le char est %s", char_a_trouver, message, p_char);
    }
    return 0;
}
```

taille de message " hello world " = 11
la char 'o' est present dans la chaine " hello world " la fin du message apres le char est o world

Il retourne le premier
caractère trouvé si il y a des
doublons

Strcmp(string.h)

`int strcmp(chaîne 1, chaîne 2)`

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

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

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

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

Strcmp(string.h)

Illustration sur CodeBlocks

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

int main()
{
    char message[] = "hello world";

    if(strcmp(message,"hello world")== 0)
    {
        printf("\nles deux chaines sont identiques\n");
    }
    else
    {
        printf("\nles deux chaines sont differentes \n");
    }
    return 0;
}
```

les deux chaines sont identiques

Strcmp(string.h)

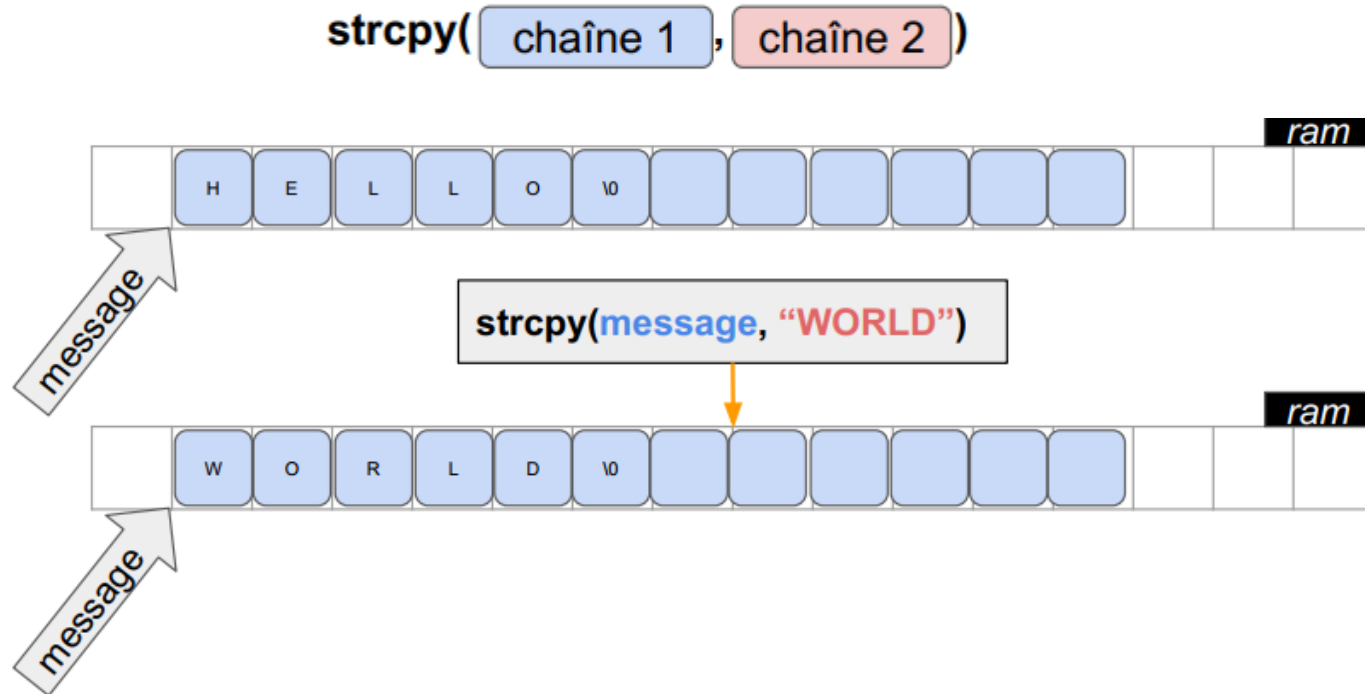
Illustration sur CodeBlocks

```
int main()
{
    char message[] = "hello world";

    if(strcmp(message,"hello SMI")== 0)
    {
        printf("\nles deux chaines sont identiques\n");
    }
    else
    {
        printf("\nles deux chaines sont differentes \n");
    }
    return 0;
}
```

les deux chaines sont differentes

Strcpy(string.h)



**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.**

Strcpy(string.h)

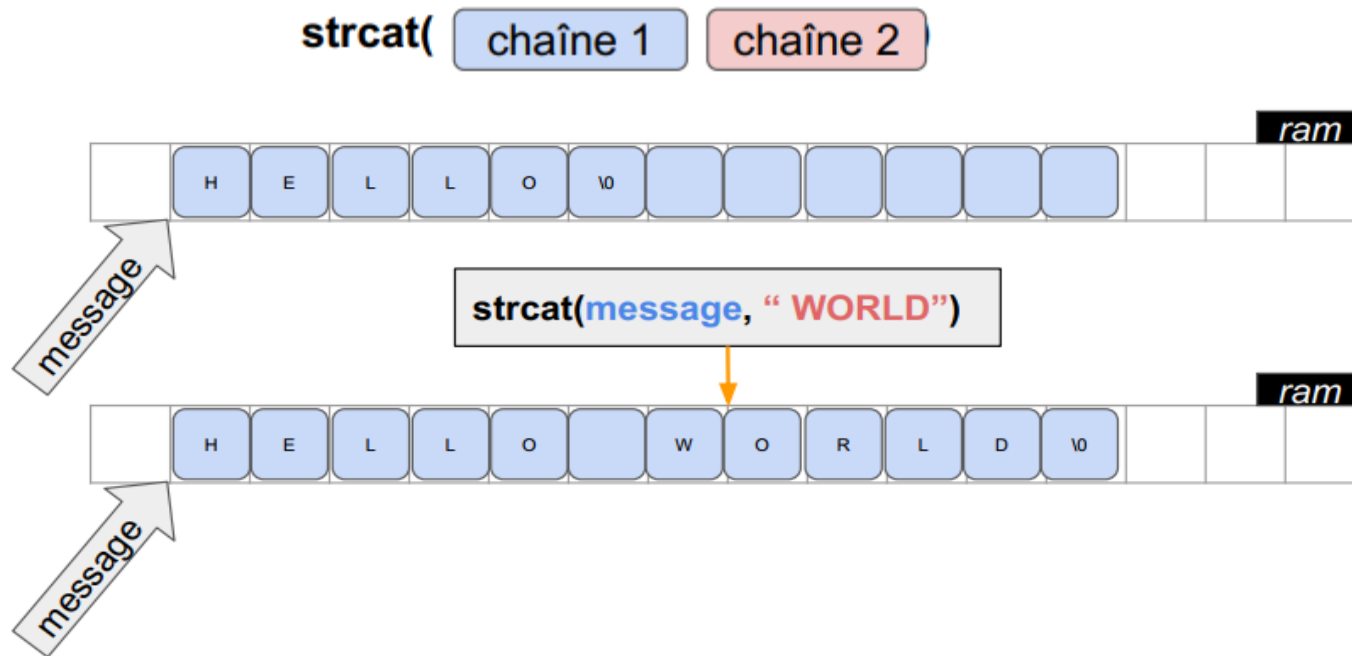
Illustration sur CodeBlocks

```
int main()
{
    char message[100] = "hello";

    printf("message = %s\n",message);
    strcpy(message,"world");
    printf("message = %s \n",message);
    return 0;
}
```

```
message = hello
message = world
```

Strcat(string.h)



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

Strcat(string.h)

Illustration sur CodeBlocks

```
int main()
{
    char message[100] = "hello";

    strcat(message, " world");
    printf("message = %s \n", message );
    return 0;
}
```

message = hello world

Strtol(string.h)

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

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

`strtol("12€", NULL, 10) => 12`

Convertit une chaîne en 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 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.

Strtol(string.h) et Strtod(string.h)

Illustration sur CodeBlocks

```
int main()
{
    char message[100] = "hello";
```

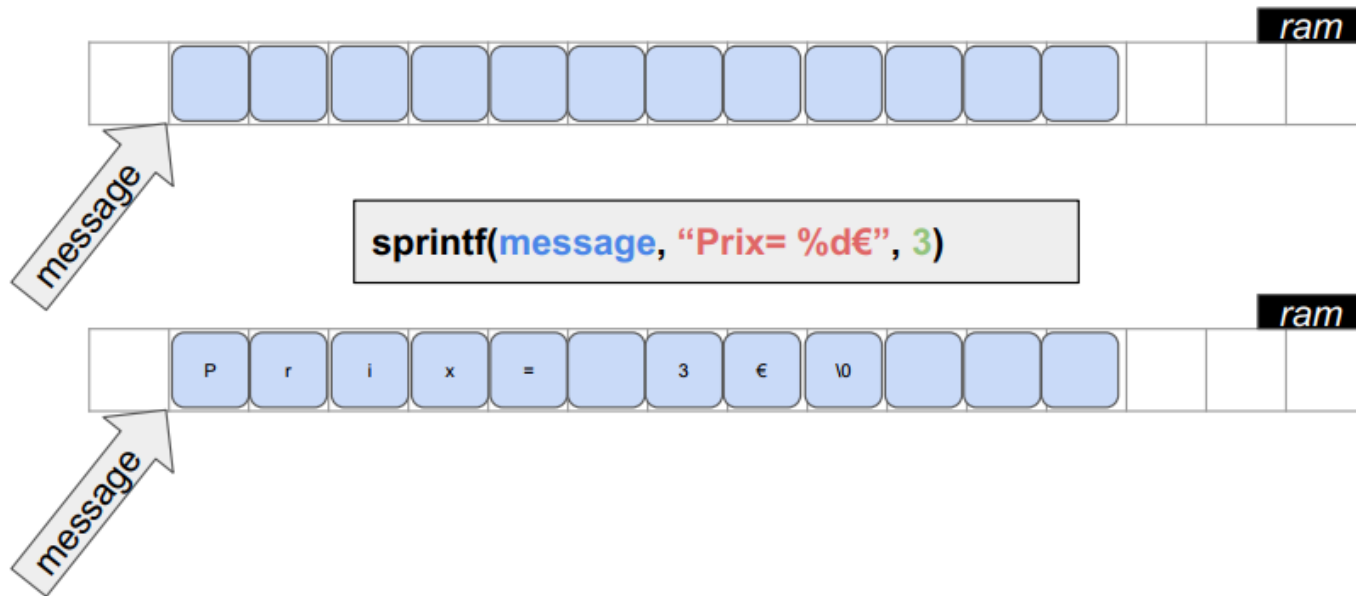
```
    printf("l entier de %s = %d \n ", "1234", strtol("1234", NULL, 10));
    printf("le float de %s = %f \n ", "123,4", strtod("123,4", NULL));
```

```
    return 0;
}
```

```
l entier de 1234 = 1234
le float de 123,4 = 123.000000
```

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).

Sprintf(stdio.h)

Illustration sur CodeBlocks

```
int main()
{
    char message[100] = "hello";
    char Nom[20];
    int Age;
    sprintf(message,"mon nom est %s et j ai %d ans ","SANAA",48);
    printf("%s",message);
    return 0;
}
```

mon nom est SANAA et j ai 48 ans

Exercice : Carte d'identité

Objectifs :

1- Créer un programme qui demande à l'utilisateur les informations suivantes et les affiche:

- Date de naissance: string au format jj/mm/aaaa
- Nom: string
- Prenom: string
- Couleur des yeux: string

donnez les informations suivantes :

Date de naissance jj/mm/aaaa :24/05/2002

Nom : KENZA

Prenom :TLEMCANI

Couleur des yeux :VERTS

vous vous appelez KENZA TLEMCANI , vous etes nee le 24/05/2002 et vos yeux sont VERTS

Aide

Il faut utiliser %s dans un scanf pour lire une string. Attention, un espace dans la saisie est considéré comme un séparateur de paramètres dans scanf.

Exercice : Carte d'identité

```
int main()
{
    char date[100];
    char nom[100];
    char prenom[100];
    char couleur[100];
    // lecture saisie utilisateur
    printf("donnez les informations suivantes : \n");
    printf("\t Date de naissance jj/mm/aaaa :");
    scanf("%s",date);
    printf("\t Nom :");
    scanf("%s",nom);
    printf("\t Prenom :");
    scanf("%s",prenom);
    printf("\t Couleur des yeux :");
    scanf("%s",couleur);
    // affichage des informations
    printf("vous vous appelez %s %s , vous etes nee le %s et vos yeux sont %s",nom,prenom,date,couleur);

    return 0;
}
```

donnez les informations suivantes :

Date de naissance jj/mm/aaaa :24/05/2002

Nom : KENZA

Prenom :TLEMCANI

Couleur des yeux :VERTS

vous vous appelez KENZA TLEMCANI , vous etes nee le 24/05/2002 et vos yeux sont VERTS

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

lecture sécurisée avec fgets

Illustration sur CodeBlocks

```
int main()
{
    char nom[100];
    printf("comment tu t'appelles ? ");
    scanf("%s",nom);

    printf("tu t'appelles %s ", nom );
    return 0;
}
```

comment tu t'appelles ? sanaa
tu t'appelles sanaa

comment tu t'appelles ? Sanaa EL FILALI
tu t'appelles Sanaa

lecture sécurisée avec fgets

Illustration sur CodeBlocks

```
int main()
{
    char nom[100];
    printf("comment tu t'appelles ? ");
    if(fgets(nom, 99, stdin) != NULL)
    {
        printf("tu ne sais rien %s ! ", nom);
    }
    return 0;
}
```

comment tu t'appelles ? Sanaa
tu ne sais rien Sanaa
!

```
int main()
{
    char nom[100];
    printf("comment tu t'appelles ? ");
    if(fgets(nom, 5, stdin) != NULL)
    {
        printf("tu ne sais rien %s ! ", nom);
    }
    return 0;
}
```

comment tu t'appelles ? SANAA
tu ne sais rien SANA !

limiter la saisie

lecture sécurisée avec fgets

Illustration sur CodeBlocks

```
int main()
{
    char prix[100];
    printf("donner un prix ? ");

    if(fgets(prix,99,stdin) != NULL)
    {
        printf("vous avez indique %f \n", strtod(prix,NULL));
    }
    return 0;
}
```

donner un prix ? 368.25
vous avez indique 368.250000

Jeux pour valider les connaissances

<https://create.kahoot.it/share/chaine-de-caractere/e8de39a6-04b4-4ef2-97d0-a39976b125d6>

Devoir à faire un jeu : le pendu

TP : les chaînes de caractères

Le pendu



Devoir à faire un jeu : le pendu

A. Enoncé :

Nous allons créer un petit jeu de type pendu. L'objectif est de deviner un mot caché (ex: -----). Pour cela le joueur peut proposer des lettres. Si une lettre est bien présente dans le mot alors elle est placée sur le mot caché (ex: --LL-). Si le joueur propose une lettre qui ne se trouve pas dans le mot alors il perd une vie.

Nous allons simplifier un peu les règles pour ce TP.

- Le mot a deviné sera écrit en dure dans le code (ex: LOL).
- Pas de mot composé (pas d'espaces).
- Le joueur ne peut que proposer des lettres une par une (pas de proposition du mot entier).
- Le joueur commence avec 10 points de vie.

```
...  
Proposer une lettre > P  
Non la lettre 'P' n'est pas présente dans le mot "---", il vous reste 9 vies  
Proposer une lettre > L  
Oui la lettre 'L' est bien présente dans le mot "L-L"  
Proposer une lettre > O  
Bravos, vous avez trouve le mot "LOL" et il vous reste 9 vies  
...
```

Devoir à faire un jeu : le pendu

A. Préparatifs

1- Créer un nouveau projet C du nom de "tp_les_chaines_de_caracteres ".

A. Instructions

1. Créer les variables et constantes nécessaires au programme (mot_secret, saisie_utilisateur, ...).
2. Créer la boucle de jeu
 - 2.1- Demander au joueur de faire une proposition de lettre (attention, utiliser la commande "fflush(stdin);" avant le scanf pour ne pas risquer de lire des restes entré au clavier).
 - 2.2- Si la proposition est bonne remplacer les '-' correspondants par la lettre. Sinon faire perdre une vie au joueur.
- 3- Gérer la fin de partie. Victoire: Cas où le joueur a trouvé toutes les lettres ou Défaite: cas où le joueur n'a plus de vies

Devoir à faire un jeu : le pendu

A. Pour aller plus loin !

Offrir la possibilité, au joueur, de

soit donner une lettre,

soit proposer directement le mot entier si il pense avoir deviné le bon mot.

Devoir à faire

un jeu : le pendu

De l'Aide..

1- Il faut créer les variables suivantes

- int nb_vie=10, char saisie_utilisateur
- le tableau qui contient la chaîne secrète : char mot_secret[] = "PROGRAMMATION";
- la taille de la chaîne secrète : int taille_mot = strlen(mot_secret);
- le tableau des char trouvés : char mot_trouve[taille_mot+1]; (Attention au + 1 dans la taille du Tableau pour avoir une case pour le '\0')
- Enfin il faut initialiser ce tableau avec des '-' et ne pas oublier le '\0' dans la dernière case

2.1- Utiliser fflush(stdin) puis un simple scanf car ici, il faut lire un simple char (attention au & devant votre variable char dans le scanf)

2.2- Il faut parcourir le tableau contenant le mot secret et vérifier pour chaque case si la saisie utilisateur est identique. Si le char est présent dans le mot secret, on l'ajoute au tableau "mot_trouve" sinon on fait perdre une vie.

3- Il faut faire une boucle do while et vérifier que le nombre de points de vie est supérieur à 0. Pour la victoire, il faut comparer la chaîne trouvée avec la chaîne du mot secret via strcmp. si elles sont identique c'est gagné ! il ne faut pas oublier de forcer la sortie de la boucle de jeu via un break;

A. Il faut remplacer la saisie utilisateur par un tableau de char et utiliser la fonction fgets au lieu du scanf. Ainsi il est possible, pour l'utilisateur, d'écrire soit un char soit un string.

Il faut séparer le cas où l'utilisateur rentre une lettre et le cas où il en rentre plusieurs (proposition d'un mot)

