

```

// cette fonction est une implémentation de la fonction strlcat en C
// ASTUCE : man strlcat
// la fonction strlcat copie et concatène le chaine d'input (src) dans la chaine de destination (dest)
// si le tampon de destination, limité par sa taille, n'est pas assez large pour contenir la copie,
// la chaine résultante sera tronquée (mais elle sera dans tous les cas terminée par '\0')

// la valeur de retour est la longueur totale de la chaine dest (après concaténation)
// que la fonction a tenté de créer (comme si la troncature ne s'était pas produite, si tel est le cas)
// ce qui correspond à la taille de la chaine de caractères dest + la chaine de caractères src

// size (nb dans notre fonction ft_strlcat) correspond à la taille totale du tampon
// (du tableau de caractères)
// incluant le caractère nul de fin de chaine
// REMARQUE : cette taille peut être calculée avec l'opérateur sizeof() sur dest
// elle correspond au nombre d'octets que le tampon peut contenir
// (en incluant le caractère de fin de chaine)
// ATTENTION : cette taille correspond à la taille totale du tableau de caractères
// par exemple, avec :
// char dest[14];
// size vaudra 14 (même s'il contient déjà des caractères)

// pour cela :
// - on fait avancer le pointeur dest jusqu'à la fin de la chaine de caractères (jusqu'à atteindre '\0')
// tout en comptant la taille de celle-ci au fur et à mesure (elle sera stockée dans dest_len,
// pour calculer la valeur de retour, qui sera dest_len + src_len, mais aussi pour vérifier
// que la taille totale du tampon ne sera pas dépassée (voir plus bas))

// - on compte la taille de la chaine de caractères src (elle sera stockée dans src_len,
// uniquement pour calculer la valeur de retour, qui sera dest_len + src_len)

// - on initialise un compteur i à 0
// - tant que la fin de la chaine src n'est pas atteinte
// et que la largeur totale de la chaine de destination (en cours de construction (concaténation)),
// qui vaut en fait dest_len + i, i étant incrémenté à chaque caractère de src ajouté à dest) est inférieure à nb
// (à la taille totale du tampon) :
//   - on ajoute le caractère pointé par src à l'emplacement en cours de dest
//   (remarque : la 1ère itération remplace le caractère de fin de chaine de dest )
//   - on fait avancer le pointeur dest

```

```
        // - on fait avancer le pointeur src
        // - on incrémente le compteur i
// une fois sorti de la boucle :
// - on ajoute le caractère de fin de chaîne à dest
// (remarque : si aucune concaténation n'a été effectuée,
// car src est vide, ou car nb (le tampon dest) est trop petit
// *dest = '\0' remplace alors le '\0' original de dest par '\0'
// cela est donc bien valide)

// - on retourne la longueur totale de la chaîne dest (après concaténation) :
// dest_len + src_len
```

```
unsigned int    ft_strlcat(char *dest, char *src, unsigned int nb)
{
    // longueur de dest
    unsigned int    dest_len;
    // longueur de src
    unsigned int    src_len;
    // compteur du nombre de caractères concaténés à dest
    unsigned int    i;

    // initialisation de la taille de dest à 0
    dest_len = 0;
    // tant que la fin de la chaîne dest n'est pas atteinte
    while (*dest != '\0')
    {
        // on fait avancer le pointeur dest
        dest++;
        // on incrémente le compteur dest_len
        dest_len++;
    }
    // initialisation de la taille de src à 0
    src_len = 0;
    // tant que la fin de la chaîne src n'est pas atteinte
    while (src[src_len] != '\0')
        // on incrémente le compteur src_len
        src_len++;
    // on initialise i à 0
```

```
i = 0;

// tant que la fin de la chaine src n'est pas atteinte
// et que la largeur totale de la chaine de destination est inférieure à nb
// (à la taille totale du tampon)
while ((*src != '\0') && (dest_len + i < nb))
{
    // on ajoute le caractère pointé par src à l'emplacement en cours de dest
    *dest = *src;
    // on fait avancer le pointeur dest
    dest++;
    // on fait avancer le pointeur src
    src++;
    // on incrémente le compteur i
    i++;
}
// on ajoute le caractère de fin de chaine à dest
*dest = '\0';
// on retourne la longueur totale de la chaine dest (après concaténation)
return (dest_len + src_len);
}
```

```
#include "ft_strcpy.h"
#include "ft_strlcat.h"
#include "ft_putnbr.h"

int    main(void)
{
    char        dest[14];
    char        src[9];
    unsigned int    nb;
    unsigned int    res;

    ft_strcpy(dest, "Hello");
    ft_strcpy(src, " World !");
    nb = sizeof(dest);
    res = ft_strlcat(dest, src, nb);
    ft_putnbr(res);
    return (0);
}

// RESULTAT : 13
```