

```

// cette fonction est une implémentation de la fonction strncmp en C
// ASTUCE : man strncmp

// la fonction strncmp est similaire à strcmp (voir ex00)
// la seule différence est que strncmp compare seulement (au plus) les n premiers octets de s1 et de s2
// (si n est plus grand que la taille de s1 ou de s2, la comparaison s'arrêtera à la fin
// de la chaîne la plus courte des deux (comme strcmp))

// idem que pour ft_strcmp, mais :
// on ajoute juste un compteur i
// de type unsigned int (de même type que n, pour pouvoir les comparer)
// qu'on initialisera à 0
// il sera incrémenté à chaque tour de boucle, en même temps que s1 et s2

// la boucle doit s'arrêter dès que i atteint n (avant que la taille ne soit dépassée)
// on ajoute donc la condition (i < n) dans le while

// REMARQUE :
// on revient à la ligne et on ajoute une tabulation pour chacune des conditions du while
// pour plus de lisibilité et pour respecter la norme
// (pour ne pas dépasser 80 colonnes pour cette ligne)
// IMPORTANT :
// chacune des lignes doit commencer par un opérateur (&& dans ce cas)

int    ft_strncmp(char *s1, char *s2, unsigned int n)
{
    // on déclare i de type unsigned int (entier positif)
    unsigned int    i;

    // on initialise i à 0
    i = 0;

    // tant que i est inférieur à n
    // (tant que la taille n'a pas été dépassée)
    while ((i < n)
           // et que l'une des chaînes n'est pas terminée
           && (*s1 && *s2)
           // et que les caractères en cours sont égaux

```

```

        && ((unsigned char)*s1 == (unsigned char)*s2))
    {
        // on incrémente s1, s2 et le compteur i
        s1++;
        s2++;
        i++;
    }
    // on retourne la différence entre les caractères de s1 et de s2
    return ((unsigned char)*s1 - (unsigned char)*s2);
}

#include "ft_strcpy.h"
#include "ft_strncmp.h"
#include "ft_putnbr.h"

int    main(void)
{
    char        s1[10];
    char        s2[10];
    unsigned int    n;
    int          res;

    ft_strcpy(s1, "Bonjour");
    ft_strcpy(s2, "Salut");
    n = 2;
    res = ft_strncmp(s1, s2, n);
    ft_putnbr(res);
    return (0);
}

```