

```

// cette fonction met en majuscule chaque lettre de
// la chaine de caractères passée en argument
// elle renvoie str (la chaine de caractères) après sa transformation éventuelle

// l'unique paramètre de cette fonction est la chaine de caractères (str)

char    *ft_strupcase(char *str)
{
    // on déclare une variable i de type int
    // puis on l'initialise à 0
    // elle servira d'indice pour parcourir un à un
    // les caractères de la chaine str
    int    i;

    i = 0;

    // tant que la chaine de caractère n'est pas terminée
    // (tant que le caractère de fin de chaine n'a pas encore été atteint)
    // (ou si la chaine de caractères n'est pas vide !)
    while (str[i] != '\0')
    {

        // on vérifie si le caractère courant, indiqué par str[i],
        // est une lettre minuscule
        // (caractère se trouvant entre a et z)
        if (str[i] >= 'a' && str[i] <= 'z')

            // ASTUCE : man ascii
            // le code décimal pour a est 97
            // celui pour A est 65
            // les codes décimaux des caractères de a à z se suivent
            // (de 65 à 90)
            // ceux des caractères de A à Z aussi
            // (de 97 à 122)
            // 97 - 65 = 32
            // donc, pour passer un caractère minuscule en majuscule
            // on soustrait 32 à son code décimal
            str[i] = str[i] - 32;
    }
}

```

```

        // la boucle passe au caractère suivant
        // en incrémentant i de 1
        i++;
    }

    // on retourne la chaîne de caractères
    return (str);
}

#include "ft_putstr.h"
#include "ft_strupcase.h"
#include <unistd.h>

int    main(void)
{

    // tableau contenant 26 caractères minuscules
    // (de a à z)
    // + le caractère de fin de chaîne de caractère '\0'
    // (voir plus bas)
    char    str[27];

    // on déclare la variable c de type char
    // qui contiendra les caractères de 'a' à 'z'
    char    c;

    // on déclare une variable i de type int
    // qu'on initialisera à 0 (voir plus bas)
    // elle servira d'indice pour parcourir un à un
    // les caractères de la chaîne str
    int     i;

    // on initialise la variable c avec le caractère 'a'
    c = 'a';
    i = 0;

    // pour construire une chaîne de caractères de a à z

```

```

// on utilise une boucle while pour répéter l'ajout du caractère
// contenu dans la variable c dans le tableau str à l'indice i
// et leur indentation (passage à la lettre suivante et à l'élément suivant du tableau)
// tant que le caractère à afficher est inférieur ou égal à 'z'
while (c <= 'z')
{
    // on stocke le caractère en cours (c) dans la chaine de caractères (str) à l'indice en cours (i)
    str[i] = c;

    // on incrémente l'indice i
    i++;

    // on incrémente le code décimal du caractère c
    c++;
}

// on ajoute le caractère de fin de chaine à la fin pour terminer la chaine de caractères
str[i] = '\0';

// on affiche la chaine de caractères str (lettres minuscules de a à z)
ft_putstr(str);

// on saute une ligne
write(1, "\n", 1);

// on met en majuscule chaque lettre de la chaine de caractères
ft_strupcase(str);

// on affiche la chaine de caractères str (lettres majuscules de A à Z)
ft_putstr(str);

// on saute une ligne
write(1, "\n", 1);
return (0);
}

```