

```

// la fonction ft_ft prend en argument
// un pointeur (*) vers un entier (int)
// ce pointeur est appelé nbr

// il pointe vers l'adresse de la variable de type int passée en paramètre
// il contiendra donc cette adresse
// ATTENTION : c'est bien l'adresse de la variable de type int qui doit être
// passée en paramètre, et non pas directement la variable

void    ft_ft(int *nbr)
{
    // ATTENTION : ici, *nbr est une "lvalue"
    // car elle se trouve à gauche (left) de l'opérande
    // d'affectation =
    // * est donc ici l'opérateur de déréférencement
    // cela signifie :
    // valeur située à l'adresse fournie par le pointeur nbr
    // cette instruction modifie la valeur de la variable pointée par nbr
    // (la valeur de la variable se trouvant à l'adresse contenue dans *nbr,
    // donc à l'adresse passée en paramètre de la fonction)
    // en lui assignant la valeur 42
    *nbr = 42;
}

// main.c :
#include "ft_ft.h"
#include <unistd.h>

void    ft_write_numbers(int n);
void    ft_write_number(int n);

int     main (void)
{
    // on définit une variable de type int appelée n
    int    n;

    // on exécute ft_ft en lui passant l'adresse
    // de la variable n de type int

```

```

    // & signifie "adresse de "
    ft_ft(&n);

    // pour afficher notre nombre

    if (n == -2147483648)
    {
        write(1, "-2147483648", 11);
        return (0);
    }
    if (n < 0)
    {
        write(1, "-", 1);
        n = -n;
    }
    ft_write_numbers(n);
    return (0);
}

void ft_write_numbers(int n)
{
    if (n >= 10)
        ft_write_numbers(n / 10);
    ft_write_number(n % 10);
}

void ft_write_number(int n)
{
    char    c;

    c = n + '0';
    write(1, &c, 1);
}

```