

```

// la fonction ft_swap prend en argument
// les adresses de deux variables de type entier
// (int *a et int *b)

// les adresses passées en paramètre de la fonction
// seront donc stockées dans les pointeurs a et b

// cette fonction pourra alors modifier la valeur
// des variables originales pointées par ces adresses

void    ft_swap(int *a, int *b)
{
    // on déclare une variable temporaire de type entier
    // elle servira à stocker temporairement
    // la valeur de l'une des variables pendant l'échange
    int    temp;

    // on stocke la valeur de la variable
    // pointée par le pointeur a (*a)
    // dans temp
    // (pour conserver cette valeur avant de la remplacer)
    // SINON, EN REMPLACANT DIRECTEMENT LA VALEUR DE LA VARIABLE
    // POINTEE PAR a PAR CELLE DE LA VARIABLE POINTEE PAR b
    // LA VALEUR DE LA VARIABLE POINTEE PAR a SERAIT ECRASEE
    // PAR CELLE DE b
    temp = *a;

    // on assigne la valeur de la variable pointée par b (*b)
    // à la variable pointée par a (*a)

    // ATTENTION :
    // *a signifie "variable pointée par a" si c'est une lvalue
    // (à gauche de l'opérateur d'affectation =)
    // (comme pour la ligne suivante)
    // MAIS
    // *a signifie "valeur de la variable pointée par a" si
    // ce n'est pas une lvalue
    // (à droite de l'opérateur d'affectation =)

```

```

// (comme pour la ligne précédente)

// TOUT COMME, POUR a = a;
// a à gauche de l'opérateur d'affectation signifie :
// variable a
// et a à droite de l'opérateur d'affectation signifie :
// valeur de la variable a

*a = *b;

// on affecte à la variable pointée par b (*b)
// la valeur de la variable temp
// (qui correspond à la valeur originale de la variable pointée par a)
*b = temp;
}

```

// main.c :

```

#include "ft_putnbr.h"
#include "ft_swap.h"
#include <unistd.h>

int    main(void)
{
    // on déclare une variable entière i
    int    i;

    // on déclare une variable entière j
    int    j;

    // on initialise i à 1
    i = 1;

    // on initialise j à 2
    j = 2;

    // on affiche la valeur de i
    // résultat : 1

```

```
ft_putnbr(i);

// on affiche un espace
write(1, " ", 1);

// on affiche la valeur de j
// résultat : 2
ft_putnbr(j);

// on retourne à la ligne
// (pour séparer les valeurs avant et après l'échange)
write(1, "\n", 1);

// on échange les valeurs de i et de j
// en passant en paramètre
// l'adresse de i et l'adresse de j
ft_swap(&i, &j);

// on affiche la valeur de i après l'échange
// résultat : 2
ft_putnbr(i);

// on affiche un espace
write(1, " ", 1);

// on affiche la valeur de j après l'échange
ft_putnbr(j);

// on retourne à la ligne
write(1, "\n", 1);

// on indique que le programme s'est terminé avec succès
return (0);
}

// RESULTAT :
// 1 2
// 2 1
```