```c
// cette fonction itérative renvoie le résultat du nombre nb
// mis à la puissance power
// une puissance inférieure à 0 renverra 0
// 0 puissance 0 renverra 1 (comme n'importe quel nombre
// positif puissance 0)
// la fonction ne gère pas les int overflow

int     ft_iterative_power(int nb, int power)
{
        // variable pour stocker le résultat
        int     res;

        // si la puissance est inférieure à 0
        if (power < 0)
                // on retourne 0
                return (0);
        // sinon, si la puissance est égale à 0
        else if (power == 0)
                // on renvoie 1
                // (car nb puissance 0 = 1)
                return (1);

        // si la puissance est 1
        // on renvoie directement le nombre
        // (car nb puissance 1 = nb)
        // POUR PLUS DE LISIBILITE
        // CAR SI power == 1
        // ON NE RENTRERA PAS DANS LA BOUCLE while (power >= 2)
        // ET res SERA DIRECTEMENT RETOURNÉ
        else if (power == 1)
                return (nb);

        // on initialise res à nb
        res = nb;

        // tant que power est supérieur à 1
        // (il sera décrémenté jusqu'à 2)
        // EXEMPLE : pour power = 4
```

```
        // power vaudra 4, puis 3, puis 2

        // EXPLICATIONS POUR nb = 5 et power = 4
        // res = 5
        // 1) res = res * nb = 5 * 5 = 25
        // 1') power-- = power - 1 = 4 - 1 = 3
        // 2) res = res * nb = 25 * 5 = 125
        // 2') power-- = power - 1 = 3 - 1 = 2
        // 3) res = res * nb = 125 * 5 = 625
        // 3') power-- = power - 1 = 2 - 1 = 1
        // FIN DE LA BOUCLE
        // 5 ^ 4 est bien égal à 625
        while (power > 1)
        {
                // on multiplie le résultat précédent par nb
                res = res * nb;

                // on décrémente power de 1
                power--;
        }
        // on retourne le résultat
        return (res);
}

// RESULTAT : 625
#include "ft_putnbr.h"
#include "ft_iterative_power.h"

int     main(void)
{
        int     nb;
        int     power;
        int     res;

        nb = 5;
        power = 4;
        res = ft_iterative_power(nb, power);
        ft_putnbr(res);
```

```
        return (0);
}
```