

```
// ce programme (composé d'une fonction main)
// affiche les arguments reçus en ligne de commande
// (un par ligne et dans le même ordre
// que ceux reçus en ligne de commande)
// compilé avec gcc -Wall -Wextra -Werror ft_print_program_name.c
// puis exécuté avec ./a.out test1 test2 test3
// doit donc afficher test1 test2 test3 à l'écran
// (mais exécuté avec ./a.out seulement,
// ne doit rien afficher)

// pour pouvoir utiliser la fonction write
#include <unistd.h>

// fonction pour afficher la chaine de caractères
// d'un argument reçu en ligne de commande
void    ft_write_argv(char *str);

// voir ex00
int     main(int argc, char **argv)
{
    // on déclare un entier i,
    // compteur du nombre d'arguments
    // passés en ligne de commande
    int    i;

    // on initialise i à 1
    // (pas à 0, car argv[0] est le nom du programme
    // qu'on saute (on ne souhaite pas l'afficher))
    i = 1;

    // argc correspond au nombre d'arguments
    // ATTENTION : argc vaut toujours au moins 1
    // car il prend en compte le nom du programme !
    // si 1 paramètre est passé en ligne de commande,
    // il vaudra 2

    // EXEMPLE AVEC 3 ARGUMENTS, test1 test2 et test3 :
    // argc = 4
```

```

// i = 1
// 1 < 4 ? OUI
// 1)
// on affiche la chaine de caractères pointée argv[1]
// (celle du premier paramètre passé en ligne de commandes)
// test1
// i++
// i = 2
// 2 < 4 ? OUI
// 2)
// on affiche la chaine de caractères pointée argv[2]
// (celle du deuxième paramètre passé en ligne de commandes)
// test2
// i++
// i = 3
// 3 < 4 ? OUI
// 3)
// on affiche la chaine de caractères pointée argv[3]
// (celle du troisième paramètre passé en ligne de commandes)
// test3
// i++
// i = 4
// 3 < 4 ? NON
// FIN DE LA BOUCLE

while (i < argc)
{
    // on affiche la chaine de caractères en cours à l'écran
    ft_write_argv(argv[i]);

    // une fois la chaine de caractères affichée à l'écran
    // on saute une ligne
    write(1, "\n", 1);

    // on incrémente i de 1
    i++;
}

```

```
        // on retourne 0 pour indiquer que le programme s'est bien exécuté
        return (0);
    }

    // pour écrire la chaine de caractères à l'écran
    // argv[1] par exemple
    void    ft_write_argv(char *str)
    {
        // voir ex00
        // on parcourt la chaine de caractères
        while (*str != '\0')
            // on écrit le caractère en cours (pointé par str)
            // puis on incrémente str
            write(1, str++, 1);
    }
}
```