

```

// cette fonction vérifie si la chaine de caractères passée en argument
// contient uniquement des caractères affichables (imprimables)

// elle renverra 0 si la chaine contient un ou plusieurs caractères
// d'un autre type
// elle renverra 1 si elle contient uniquement des caractères affichables (imprimables)
// ou si c'est une chaine vide
// l'unique paramètre de cette fonction est la chaine de caractères (str)

// les caractères affichables (imprimables) se trouvent entre ' ' (espace) et '~' (sans les guillemets)
// (entre 32 et 126 en code ASCII décimal)

int    ft_str_is_printable(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 autre qu'un caractère affichable (imprimable)
        // (autre qu'un caractère se trouvant entre ' ' et '~')
        if (!(str[i] >= ' ' && str[i] <= '~'))

            // si le caractère courant n'est pas un caractère affichable (imprimable)
            // la fonction retourne immédiatement 0
            // car cela signifie que la chaine contient au moins un caractère
            // qui n'est pas affichable (imprimable)
            // cela permet aussi d'arrêter la fonction et de retourner le résultat
            // dès que possible !

```

```

        return (0);

        // si le caractère courant est imprimable, la boucle passe au caractère suivant
        // en incrémentant i de 1
        i++;
    }

    // si la boucle se termine sans rencontrer de caractère non imprimable, après avoir vérifié
    // tous les caractères jusqu'au caractère nul
    // la fonction retourne 1
    // elle retournera aussi 1 si aucun caractère n'a été vérifié
    // c'est-à-dire si la chaîne str est nulle
    // ce qui correspond au comportement voulu !
    return (1);
}

```

// main.c :

```

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

int    main(void)
{
    // pour stocker le retour de la fonction ft_str_is_printable
    // (0 ou 1)
    int    is_printable;

    // tableau contenant 31 caractères non imprimables
    // (entre les caractères de code décimal ASCII 1 à 31 compris)
    // + le caractère de fin de chaîne de caractère '\0'
    // (voir plus bas)
    char    str1[32];

    // tableau contenant le dernier caractère non imprimable
    // (de code décimal ASCII 127)
    // + le caractère de fin de chaîne de caractère '\0'
    // (voir plus bas)

```

```

char    str2[2];

// tableau contenant le caractère de code décimal ASCII 31
// (non imprimable) et le caractère de code décimal ASCII 32
// (imprimable)
// + le caractère de fin de chaîne de caractère '\0'
// (voir plus bas)
char    str3[3];

// compteur pour parcourir str1
int      i;

// IMPORTANT :
// on commence à 1 pour ne pas inclure le caractère nul '\0'
// sinon, la chaîne de caractères ne serait pas analysée par ft_str_is_printable !
// (str serait interprété comme terminée dès le premier caractère !)
i = 1;

// tant que i est inférieur à 32
// (pour ajouter à la chaîne de caractères les caractères de 1 à 32 non compris)
while (i < 32)
{
    // on ajoute le caractère au tableau
    // ATTENTION :
    // on indique i - 1 car i commence à 1 !
    // sinon, les caractères seraient ajoutés à partir de l'indice 1
    // (le premier caractère de la chaîne de caractères ne serait pas défini)
    str1[i - 1] = i;
    i++;
}

// on ajoute le caractère de fin de chaîne à la fin pour terminer la chaîne de caractères
str1[i] = '\0';
str2[0] = 127;
str2[1] = '\0';
str3[0] = 31;
str3[1] = 32;
str3[2] = '\0';

```

```

    is_printable = ft_str_is_printable("abcdefghijklmnopqrstuvwxyz");
    ft_putnbr(is_printable);
    write(1, "\n", 1);
    is_printable = ft_str_is_printable("ABCDEFGHIJKLMNOPQRSTUVWXYZ");
    ft_putnbr(is_printable);
    write(1, "\n", 1);
    is_printable = ft_str_is_printable("aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ");
    ft_putnbr(is_printable);
    write(1, "\n", 1);
    is_printable = ft_str_is_printable("0123456789");
    ft_putnbr(is_printable);
    write(1, "\n", 1);
    is_printable = ft_str_is_printable("a0b1c2d3e4f5g6h7i8j9");
    ft_putnbr(is_printable);
    write(1, "\n", 1);
    is_printable = ft_str_is_printable("A0B1C2D3E4F5G6H7I8J9");
    ft_putnbr(is_printable);
    write(1, "\n", 1);
    is_printable = ft_str_is_printable(str1);
    ft_putnbr(is_printable);
    write(1, "\n", 1);
    is_printable = ft_str_is_printable(str2);
    ft_putnbr(is_printable);
    write(1, "\n", 1);
    is_printable = ft_str_is_printable(str3);
    ft_putnbr(is_printable);
    write(1, "\n", 1);
    return (0);
}

```

// RESULTAT :

```

// 1
// 1
// 1
// 1
// 1
// 1
// 1
// 0

```

```
// 0
// 0
```