

```

1 // cette fonction vérifie si la chaine de caractères passée en argument
2 // contient uniquement des caractères affichables (imprimables)
3
4 // elle renverra 0 si la chaine contient un ou plusieurs caractères
5 // d'un autre type
6 // elle renverra 1 si elle contient uniquement des caractères affichables (imprimables)
7 // ou si c'est une chaine vide
8 // l'unique paramètre de cette fonction est la chaine de caractères (str)
9
10 // les caractères affichables (imprimables) se trouvent entre ' ' (espace) et '~' (sans les guillemets)
11 // (entre 32 et 126 en code ASCII décimal)
12
13 int ft_str_is_printable(char *str)
14 {
15     // on déclare une variable i de type int
16     // puis on l'initialise à 0
17     // elle servira d'indice pour parcourir un à un
18     // les caractères de la chaine str
19     int i;
20
21     i = 0;
22
23     // tant que la chaine de caractère n'est pas terminée
24     // (tant que le caractère de fin de chaine n'a pas encore été atteint)
25     // (ou si la chaine de caractères n'est pas vide !)
26     while (str[i] != '\0')
27     {
28         // on vérifie si le caractère courant, indiqué par str[i],
29         // est autre qu'un caractère affichable (imprimable)
30         // (autre qu'un caractère se trouvant entre ' ' et '~')
31         if (!(str[i] >= ' ' && str[i] <= '~'))
32
33             // si le caractère courant n'est pas un caractère affichable (imprimable)
34             // la fonction retourne immédiatement 0
35             // car cela signifie que la chaine contient au moins un caractère
36             // qui n'est pas affichable (imprimable)
37             // cela permet aussi d'arrêter la fonction et de retourner le résultat
38             // dès que possible !
39             return (0);
40         // si le caractère courant est imprimable, la boucle passe au caractère suivant
41         // en incrémentant i de 1
42         i++;
43     }
44
45     // si la boucle se termine sans rencontrer de caractère non imprimable, après avoir vérifié
46     // tous les caractères jusqu'au caractère nul
47     // la fonction retourne 1
48     // elle retournera aussi 1 si aucun caractère n'a été vérifié
49     // c'est-à-dire si la chaine str est nulle
50     // ce qui correspond au comportement voulu !
51     return (1);

```

```

52  }
53
54  // main.c :
55  #include "ft_str_is_printable.h"
56  #include "ft_putnbr.h"
57  #include <unistd.h>
58  int main(void)
59  {
60  #include "ft_str_is_printable.h"
61  #include "ft_putnbr.h"
62  #include <unistd.h>
63
64  // pour tester les caractères non imprimables
65  void ft_test_non_printable(int min, int max);
66
67  int main(void)
68  {
69      int is_printable;
70
71      ft_test_non_printable(1, 32);
72      ft_test_non_printable(127, 127);
73      ft_test_non_printable(31, 32);
74      is_printable = ft_str_is_printable("abcdefghijklmnopqrstuvwxyz");
75      ft_putnbr(is_printable);
76      write(1, "\n", 1);
77      is_printable = ft_str_is_printable("ABCDEFGHIJKLMNOPQRSTUVWXYZ");
78      ft_putnbr(is_printable);
79      write(1, "\n", 1);
80      is_printable = ft_str_is_printable(
81          "aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ");
82      ft_putnbr(is_printable);
83      write(1, "\n", 1);
84      is_printable = ft_str_is_printable("0123456789");
85      ft_putnbr(is_printable);
86      write(1, "\n", 1);
87      is_printable = ft_str_is_printable("a0b1c2d3e4f5g6h7i8j9");
88      ft_putnbr(is_printable);
89      write(1, "\n", 1);
90      is_printable = ft_str_is_printable("A0B1C2D3E4F5G6H7I8J9");
91      ft_putnbr(is_printable);
92      write(1, "\n", 1);
93      return (0);
94  }
95
96  void ft_test_non_printable(int min, int max)
97  {
98      char str[50];
99      int i;
100     int j;
101     int is_printable;
102

```

```
103     i = 0;
104     j = min;
105     while (j <= max)
106     {
107         str[i] = j;
108         i++;
109         j++;
110     }
111     str[i] = '\0';
112     is_printable = ft_str_is_printable(str);
113     ft_putnbr(is_printable);
114     write(1, "\n", 1);
115 }
```