



EXERCISES — Alphabet Checker

version #



IT IS MY JOB TO MAKE SURE YOU DO YOURS.

Copyright

This document is for internal use at EPITA ([website](#)) only.

Copyright © 2021-2022 Assistants <assistants@tickets.assistants.epita.fr>

The use of this document must abide by the following rules:

- ▷ You downloaded it from the assistants' intranet.*
- ▷ This document is strictly personal and must **not** be passed onto someone else.
- ▷ Non-compliance with these rules can lead to severe sanctions.

Contents

1	Alphabet Checker	3
1.1	Goal	3
1.2	Example	3

*<https://intra.assistants.epita.fr>

1 Alphabet Checker

Files to submit:

- check_alphabet/check_alphabet.c
- check_alphabet/check_alphabet.h

Provided files:

- check_alphabet/check_alphabet.h

Authorized headers: You are only allowed to use the functions defined in the following headers:

- errno.h
- err.h
- assert.h
- stddef.h

1.1 Goal

In this exercise, we want to check that our whole alphabet is used, it would be a shame to let some characters not being used, and so, being alone!

Write a function that takes two strings (the first one being the string to check, and the second one the alphabet) and check that every characters declared in the alphabet string is used at least once in the given string. If every characters from the alphabet are in the given string, you have to return 1, otherwise, you have to return 0. All the characters of the alphabet will always be unique.

Implement the `check_alphabet` function:

```
int check_alphabet(const char *str, const char *alphabet);
```

- The string to check will never be NULL.
- We will consider that a NULL or an empty dictionary ("") always belongs to any given string.
- The alphabet can contain any valid ASCII character, except '\0' which is used to terminate the string.

1.2 Example

```
#include <stdio.h>

#include "check_alphabet.h"

int main(void)
{
    printf("%d\n", check_alphabet("toto", NULL));
    printf("%d\n", check_alphabet("", "t"));
    printf("%d\n", check_alphabet("toto asticot", "otaisc k"));
}
```

(continues on next page)

(continued from previous page)

```
    printf("%d\n", check_alphabet("toto asticot", "ot"));

    return 0;
}
```

```
42sh$ gcc -Wall -Wextra -Werror -std=c99 -pedantic -o check_alphabet
check_alphabet.c main.c
42sh$ ./check_alphabet | cat -e
1$
0$
0$
1$
42sh$
```

It is my job to make sure you do yours.