Exercises — Check Alphabet

version **#7be580532266ed398481e31366afcc24b1950c2a**

Assistants C/Unix 2023 <assistants@tickets.assistants.epita.fr>

Copyright

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

Contents

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

File Tree

```
check_alphabet/
├── check_alphabet.c   (to submit)
└── check_alphabet.h   (to submit)
```

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

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

Compilation : Your code must compile with the following flags

- -std=c99 -pedantic -Werror -Wall -Wextra -Wvla

Main function : None

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:

```c
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.

2 Example

```c
#include <stdio.h>

#include "check_alphabet.h"

int main(void)
{
```

(continues on next page)

(continued from previous page)

```c
    printf("%d\n", check_alphabet("toto", NULL));
    printf("%d\n", check_alphabet("", "t"));
    printf("%d\n", check_alphabet("toto asticot", "otaisc k"));
    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$
```

The way is lit. The path is clear. We require only the strength to follow it.