



# EXERCISES — My strstr

---

version #7be580532266ed398481e31366afcc24b1950c2a



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

# Copyright

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

Copyright © 2022-2023 Assistants <[assistants@tickets.assistants.epita.fr](mailto: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 Goal

3

---

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

## File Tree

```
my_strstr/  
├── my_strstr.c  (to submit)  
└── my_strstr.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

Write a function that behave like `strstr(3)`. Your function must look for the first occurrence of the `needle` string within the `haystack` string, and return the index to the beginning of `needle`. If `needle` was not found, `my_strstr` must return `-1`.

```
int my_strstr(const char *haystack, const char *needle);
```

If `needle` is empty, you must return 0. The case where only `haystack` is empty will not be tested.

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