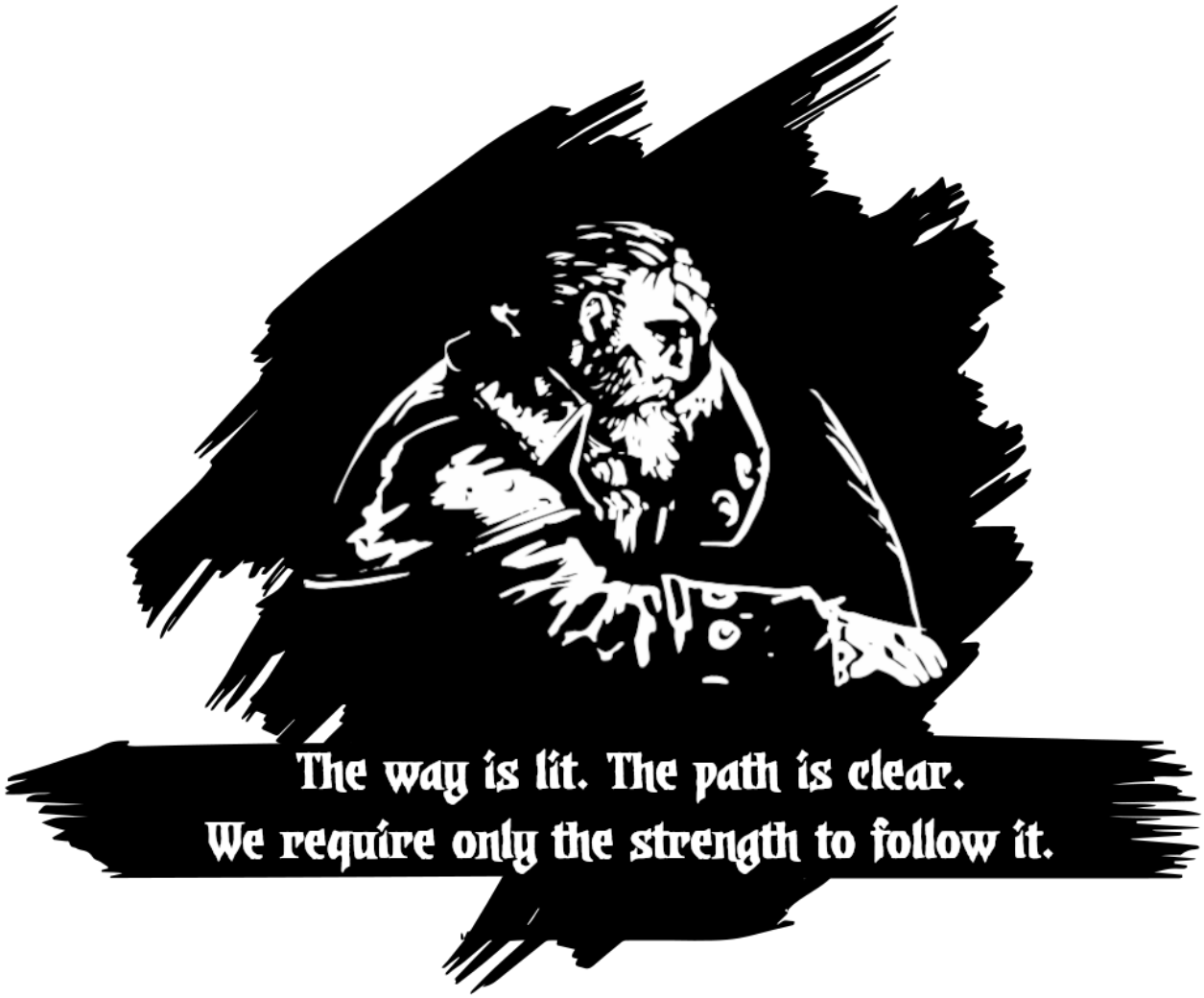
Exercises — Simple fnmatch

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

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

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

You have to implement a subset of the `fnmatch(3)` function from the standard `C` library, with the following signature.

```c
int simple_fnmatch(const char *pattern, const char *string);
```

The expected return value is `0` if there is a match between the pattern and the string, `FNM_NOMATCH` otherwise.

2 Requirements

Your function must accept and handle accurately:

- the `?` wildcard, matching any one character,
- the `*` wildcard, matching 0 or more characters,
- the escaping character `\\`.

Your function shall have the same behaviour as `fnmatch(3)` when called with an empty `flags` argument (the value `0`).

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