



EXERCISES — File Basics: Insert Line

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>

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

```
file_basics_insert_line/  
└─ insert_line.c (to submit)
```

Authorized functions : You are only allowed to use the following functions

- fopen
- fclose
- fseek
- fgetc
- getline
- fputs
- free

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 copies the content of `file_in` into `file_out` and adds `content` at the beginning of the n^{th} line. No additional character is needed between `content` and the rest of the file.

If `file_in` is too small, add as many linefeed as needed.

If any error occurs or NULL parameters, return -1. Otherwise, return the new number of lines in `file_out`.

You must open `file_in` with read-only and `file_out` with write only permission.

Note: line indices start at 0.

```
int insert_line(const char *file_in, const char *file_out, const char *content,  
               size_t n);
```

Be careful!

This exercise will be compiled with `-D_XOPEN_SOURCE=700`

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