Exercises — Assignment Operator

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

```
assignment_operator/
└── assignment_operator.c  (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 set of functions performing basic arithmetic operations on pointers to integers.

In this exercise, if an argument is `NULL`, the functions must not do anything.

1.1 Addition assignment

Write a function that takes two `int*` as parameters and stores the sum of values pointed to by both arguments in the first one.

1.1.1 Prototype

```c
void plus_equal(int *a, int *b);
```

1.2 Subtraction assignment

Write a function that takes two `int*` as parameters and stores the difference between the values pointed to by both arguments in the first one.

1.2.1 Prototype

```
void minus_equal(int *a, int *b);
```

1.3 Multiplication assignment

Write a function that takes two `int*` as parameters and stores the product of values pointed to by both arguments in the first one.

1.3.1 Prototype

```
void mult_equal(int *a, int *b);
```

1.4 Division assignment

Write a function that takes two `int*` as parameters and stores the euclidian division of values pointed by both arguments in the first one. The function must return the rest of the euclidian division.

If an argument is `NULL` or if `*b == 0`, the function must not do anything and return 0.

1.4.1 Prototype

```
int div_equal(int *a, int *b);
```

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