Exercise — Beware Overflow

version **#bfdaecd01cbf44dfbd7800b940343997d9ad7d73**

Assistants C/Unix 2025 <assistants@tickets.assistants.epita.fr>

Copyright

Contents

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

File Tree

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

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

1 Definition

An Integer *Overflow* occurs when the maximum value of an integer is reached. In C the behaviour of a signed overflow is undefined. For an unsigned overflow, the number is reduced by the modulo of the largest representable value + 1 (see below).

Example:

```c
#include <stdint.h>
#include <stdio.h>

int main(void)
{
    uint8_t a = 255;
    uint8_t b = 3;
    uint8_t c = a + b;
    printf("%u + %u = %u\n", a, b, c);
}
```

```
42sh$ gcc -pedantic -Werror -Wall -Wextra -std=c99 overflow.c -o overflow
42sh$ ./overflow
255 + 3 = 2
```

One might expect a result of 258, but the result is 2. Why?

On 8 bits there are 2^8 (256) possible values (from 0 to 255). The maximum value is 255, so:

```
a = 255
b = 3
c = (a + b) % (255 + 1)
=> c = 2
```

2 Goal

The goal of this exercise is to increase the value of the pointer `ptr` by `nmemb` elements of `size` bytes.

In order to do that you have to implement the following function:

```c
void *beware_overflow(void *ptr, size_t nmemb, size_t size);
```

In case an overflow occurs, you must return `NULL`. Otherwise, return the increased value of the pointer. The overflow must only be checked for the `nmemb` and `size` multiplication. We assume that adding the result with `ptr` never overflows.

To check this overflow you must use the gcc builtin function. The online documentation is available here, read it and choose the right function to use.

```
42sh$ gcc -Wall -Wextra -Werror -std=c99 -pedantic -o main main.c \
      beware_overflow.c
42sh$ ./main
Pointer was incremented from 0x1000 to 0x1096.
Overflow detected between 12345678904 and 12345678904.
```

Seek strength. The rest will follow.