Exercise — Page begin

version **#bfdaecd01cbf44dfbd7800b940343997d9ad7d73**

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

Copyright

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

Contents

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

File Tree

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

1 Definition

A page is a block of continuous memory of a fixed size.

2 Goal

It is sometimes useful to be able to read the metadata at the beginning of a page, and to do this you have to find that beginning.

The purpose of this exercise is to find the address of the beginning of a page, given its size and a pointer to an address in that page.

Tips

Google might be your best friend.

For this exercise, you are exceptionally allowed to use one explicit cast. Use it wisely.

Information:

- The size of a page is fixed.
- The size of a page is a power of 2.
- The *, /, and % operators must not be used.
- The variable *page_size* is bounded between 16 and 2 147 483 648.

```c
void *page_begin(void *ptr, size_t page_size);
```

3 Example

```c
#include <stdio.h>

#include "page_begin.h"

static void display_result(void *ptr, size_t page_size, void *expected_result)
{
    void *res = page_begin(ptr, page_size);

    printf("ptr: %p\n", ptr);
    printf("page_size: %lu\n", page_size);
    printf("expected_result: %p\n", expected_result);
    printf("result: %p\n", res);

    printf(expected_result == res ? "OK\n" : "KO\n");
}

int main()
{
    display_result((void *)0x1234ffea, 4096, (void *)0x1234f000);
    display_result((void *)0x1234ffea, 256, (void *)0x1234ff00);
}
```

```
42sh$ gcc -Wall -Wextra -Werror -std=c99 -pedantic page_begin.c main.c
42sh$ ./a.out
ptr: 0x1234ffea
page_size: 4096
expected_result: 0x1234f000
result: 0x1234f000
OK
ptr: 0x1234ffea
page_size: 256
expected_result: 0x1234ff00
result: 0x1234ff00
OK
```

Seek strength. The rest will follow.