Exercises — Test a Bit

version **#7be580532266ed398481e31366afcc24b1950c2a**

Assistants C/Unix 2023 <assistants@tickets.assistants.epita.fr>

Copyright

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

Contents

1 Goal 3

2 Example 3

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

File Tree

```
test_a_bit/
├── is_set.c  (to submit)
└── is_set.h  (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 the `is_set` function that returns `true` if the `n`th bit is set, `false` otherwise.

`n` is 1-based and while always be greater than zero.

```
unsigned int is_set(unsigned int value, unsigned char n);
```

2 Example

```
#include <stdio.h>

#include "is_set.h"

int main(void)
{
    printf("%d\n", is_set(24, 4));
    printf("%d\n", is_set(24, 3));

    return 0;
}
```

```
42sh$ gcc -Wall -Werror -std=c99 -Wextra -pedantic test.c is_set.c -o is_set_example
42sh$ ./is_set_example
1
0
```

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