# Exercises — my_atoi_base

version #


Assistants C/Unix 2022 <assistants@tickets.assistants.epita.fr>

## Copyright

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



## Contents

**1 my_atoi_base** **3**
- 1.1 Goal . . . 3
- 1.2 Examples . . . 3

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

# 1 my_atoi_base

**Files to submit**:

- my_atoi_base/my_atoi_base.c
- my_atoi_base/my_atoi_base.h

**Main function:** None

**Authorized headers:** You are only allowed to use the functions defined in the following headers:

- err.h
- assert.h
- errno.h
- stddef.h

## 1.1 Goal

You have to implement the following function:

```
int my_atoi_base(const char *str, const char *base);
```

This function must have the same behavior as the `atoi(3)` function, but in a specified base.

`str` is a string and represents a number in the base `base`. `str` must be converted into the associated decimal value.

## 1.2 Examples

```
my_atoi_base("ff", "0123456789abcdef");
```

must return the value 255.

```
my_atoi_base("-ff", "0123456789abcdef");
```

must return the value -255.

```
my_atoi_base("77", "01234567");
```

must return the value 63.

```
my_atoi_base("WQWW", "QW");
```

must return the value 11.

If one of the digits of the number to convert is not included in the given base, the result must be zero.

The `str` string must follow the following format in specified order:

- A possibly empty sequence of whitespace characters that will be discarded

- An optional single plus or minus sign
- A sequence of digits included in `base` string

If `str` does not match this format, you must return 0.

The string `base` always represents a valid base. If the `str` string is empty, you must return 0 (zero).

*It is my job to make sure you do yours.*