Exercises — The Sieve of Eratosthenes (Advanced)

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

```
sieve_eratosthenes_advanced/
├── Makefile  (to submit)
└── sieve.c  (to submit)
```

Makefile

- all: Generate sieve.o

Authorized functions : You are only allowed to use the following functions

- printf(3)
- malloc(3)
- free(3)
- calloc(3)

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

The goal here is to create a sieve of Eratosthenes. It is a simple iterative sieve for finding prime numbers.

It works with an array of numbers from `2` to `n` (`n` excluded). The first number, `2`, is identified as a prime number. Then, all multiples of `2` are marked. When encountering a new unmarked number, it is identified as a prime number, and its multiples are marked.

You must write a function taking an integer `n` and printing the number of prime numbers from `2` to `n`, excluded.

```
void sieve(int n);
```

If `n` is equal or lower than `2`, the function will not print anything. Your function will have to be optimized as it will be tested with large values.

2 Examples

You can use a `main` to test your program, but it must not be included in your submission. Calling the `sieve` function with the first program argument as parameter, will produce the following output:

```
42sh$ ./sieve 11 | cat -e
4$
42sh$ ./sieve 0 | cat -e
42sh$ ./sieve 1000000 | cat -e
78498$
```

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