Documenting the Interface

Your programs should have two kinds of documentation. **Implementation comments** help other programmers understand the workings of your code, and the algorithms that you used. These should be added to your implementation file. I usually use single-line comments for these.

User-facing documentation, that needed by the users of your function, is generated by a documentation tool named **Doxygen**. Doxygen is used to generate HTML, PDF, LaTex, and other kinds user-facing documentation. You can see an example of the generated documentation here at the SFML Docs Web site.

Each comment starts with a `/**` token, and ends with a `*/` token. The individual elements begin with keywords, called **tags**. Each header file should begin with a **file comment** that contains these common elements.

> `@file` – Name of the file. Required if you are going to document functions, global variables and constants. **Always use this.**
>
> `@author` – your name. or ID (e.g. sgilbert)
>
> `@date` – date it was created (can be general, like: CS 150 S'25, MWAM)
>
> `@version` – version information about the library (optional)

Function Comments

Each function should also be documented. Place your function comments **before** the header line of the function. Here are the tags to use.

> Start your comment block with a single line ending in a period. This is the brief description of the function.
>
> `@param` – the name and description of every parameter to your function.
>
> `@return` – if the function returns a value, you should add this tag.
>
> `@code` (optional) – a **block** of code that will be syntax highlighted in the generated documentation. This block must end with a `@endcode` tag.

Here's the header file you're working on with the **first** function documented.

```
#ifnfef DIGITS_H
#define DIGITS_H
/**
  * @file digits.h
  * @author sgilbert (Stephen Gilbert)
  * @date CS 150 Spring 2025 SAT AM
*/

/**
  * Find the first digit of any integer n.
  * @param n the integer to check. May be negative.
  * @return the first digit of the integer n.
  * @code
  * int a = firstDigit(215);  // set a to 2
  * int b = firstDigit(-79);  // set b to 7, not -7
  * @endcode
*/
int firstDigit(int n);
#endif
```

Now, go and document the remaining functions on your own.