The **exit()** and **return** statements are both used to terminate a program or a function, but they have different purposes and behaviors. Here's the key difference:

**1. exit()**

* **Purpose**: Terminates the entire program immediately.
* **Scope**: Exits the program regardless of where it is called (inside main() or other functions).
* **Defined In**: <stdlib.h>.
* **Behavior**:
  + Closes all open files.
  + Performs cleanup (e.g., flushing buffers, calling functions registered with atexit()).
  + Can return a status code to the operating system (exit(0) for success, non-zero for error).
* **Use Case**: When you want to end the program from anywhere, such as in error handling or when specific conditions are met.

**2. return**

* **Purpose**: Exits the current function and optionally provides a return value.
* **Scope**: Exits only the function in which it is called.
* **Defined In**: Built into the language (not dependent on a library).
* **Behavior**:
  + Transfers control back to the caller function.
  + In main(), return can also terminate the program, similar to exit(), but is preferred for indicating the program's status.
* **Use Case**: When you want to return control and/or data to the calling function or end the main() function.