To compile kilo.c, run cc kilo.c -o kilo in your shell. If no errors occur, this will produce an executable named kilo. -o stands for “output”, and specifies that the output executable should be named kilo.

To run kilo, type ./kilo in your shell and press Enter. The program doesn’t print any output, but you can check its exit status (the value main() returns) by running echo $?, which should print 0.

makefile

Copy code

# Makefile for compiling C++ project

# Variables

CC = g++

CFLAGS = -Wall -Wextra -pedantic -std=c++11

# Target to build

my\_program: main.cpp

$(CC) $(CFLAGS) main.cpp -o my\_program

**Explanation:**

* **CC = g++**: This specifies that we are using the g++ compiler (instead of cc for C programs).
* **CFLAGS**: These are the compiler flags used when compiling. Here's what they mean:
  + **-Wall**: Enables all common warnings.
  + **-Wextra**: Enables even more warning flags, catching possible issues.
  + **-pedantic**: Enforces strict adherence to the C++ standard.
  + **-std=c++11**: Specifies the C++ version. You can change this to -std=c++14, -std=c++17, etc., based on the version of C++ you want to use.
* **my\_program: main.cpp**: This defines the target (my\_program) and the source file (main.cpp) required to build it.
* The next line is the actual compilation command, indented using a **tab** (important for Makefile).