Chapter 9:
Epsilon Test

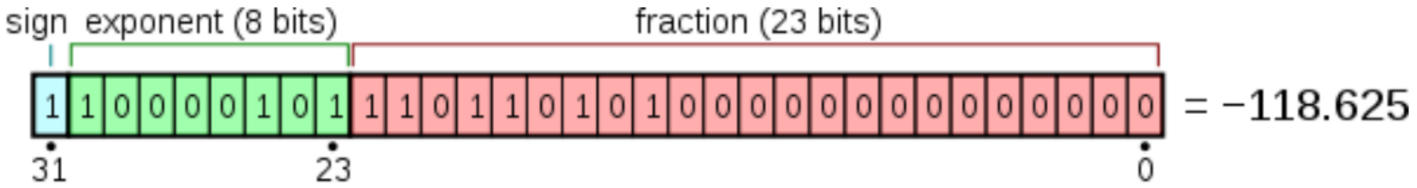
Mijin An
meeeeejin@gmail.com

Why Floating-Point Numbers May Lose Precision

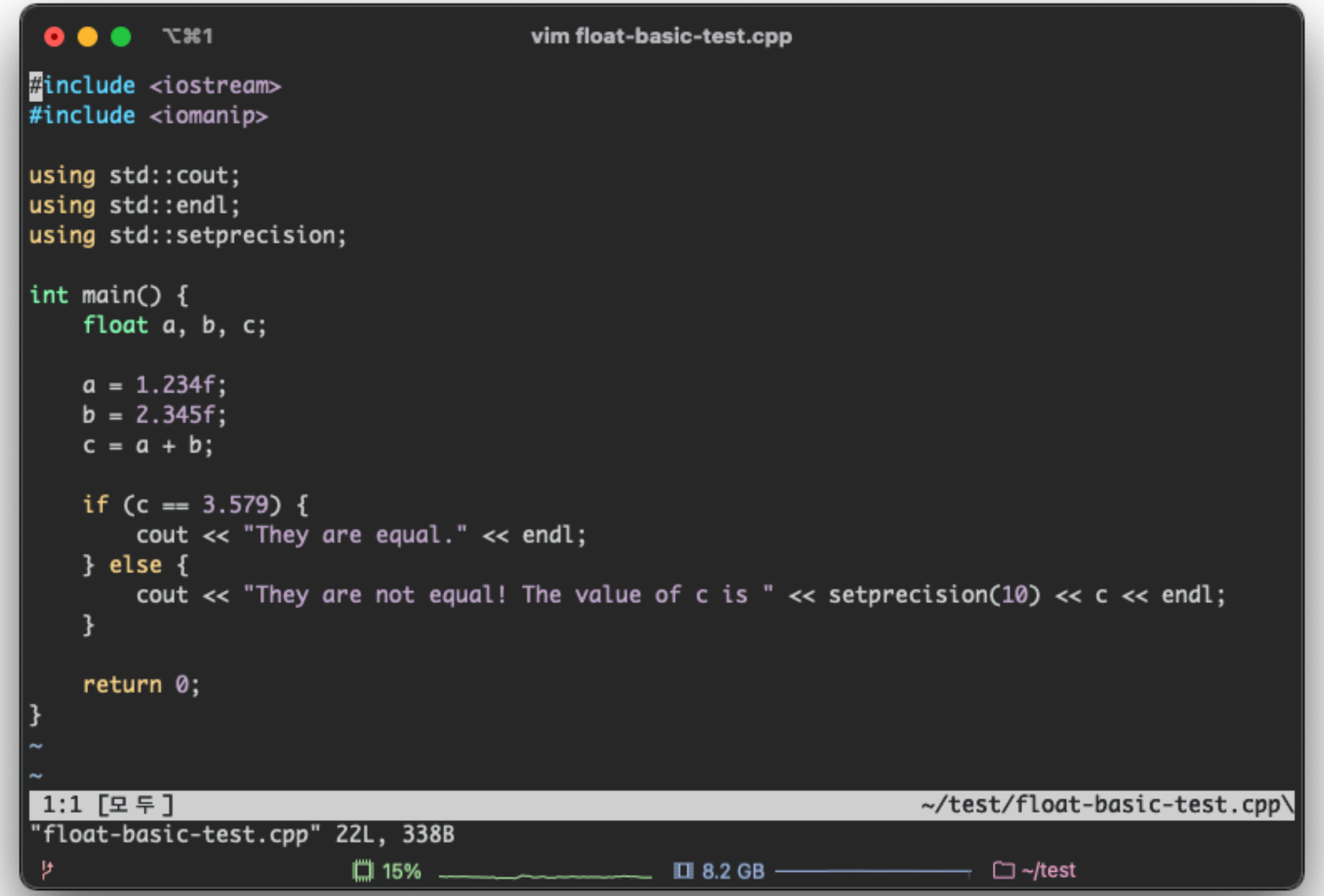
- Floating-point decimal values generally **do not have an exact binary representation**
  - A side effect of **how the CPU represents floating point data**
  - **Loss of precision** → Unexpected results

- To resolve the behavior:
  1. Ensure that the value is greater or less than what is needed
  2. Use a Binary Coded Decimal (BCD) library

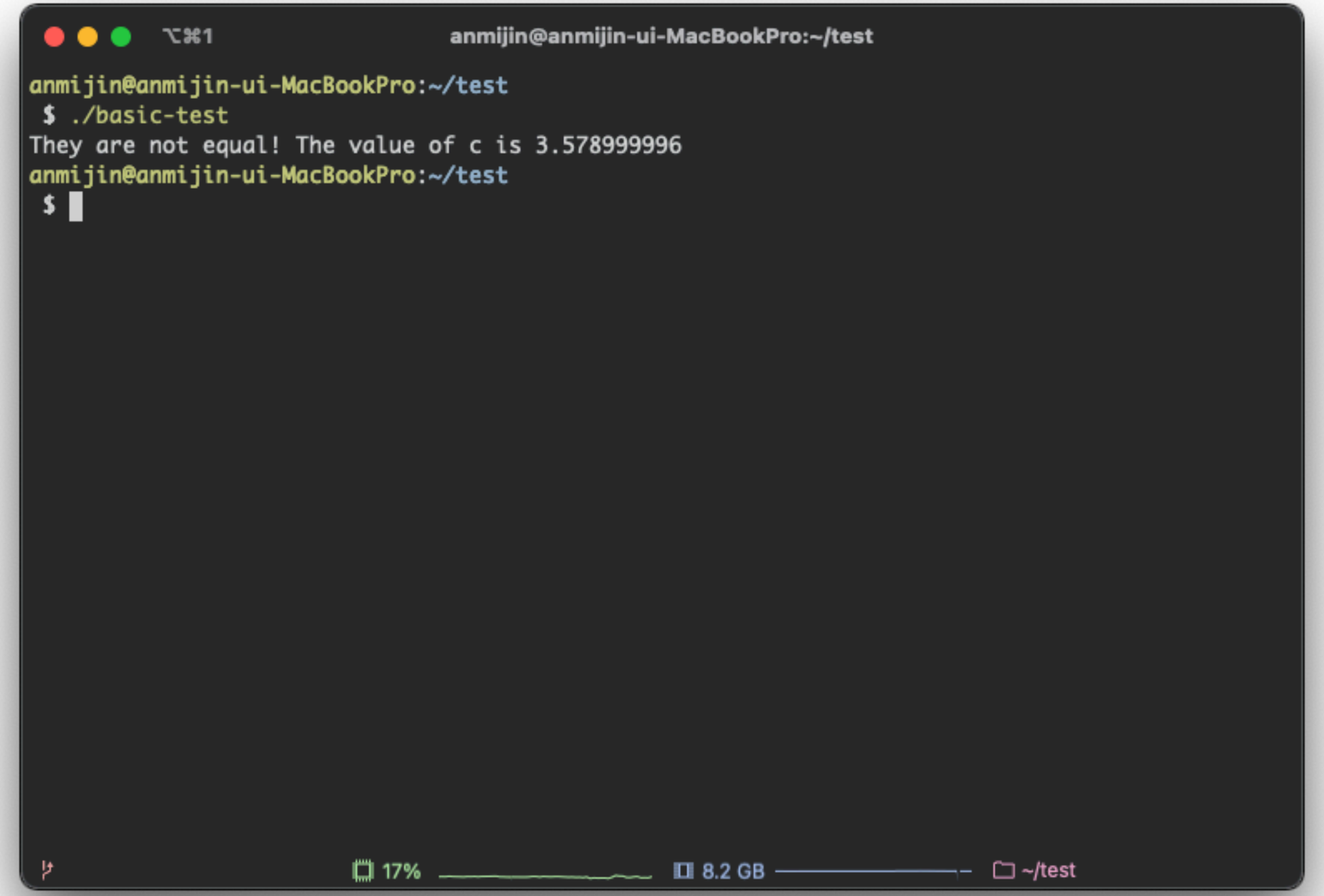
- A technical standard for floating-point arithmetic: **IEEE 754**



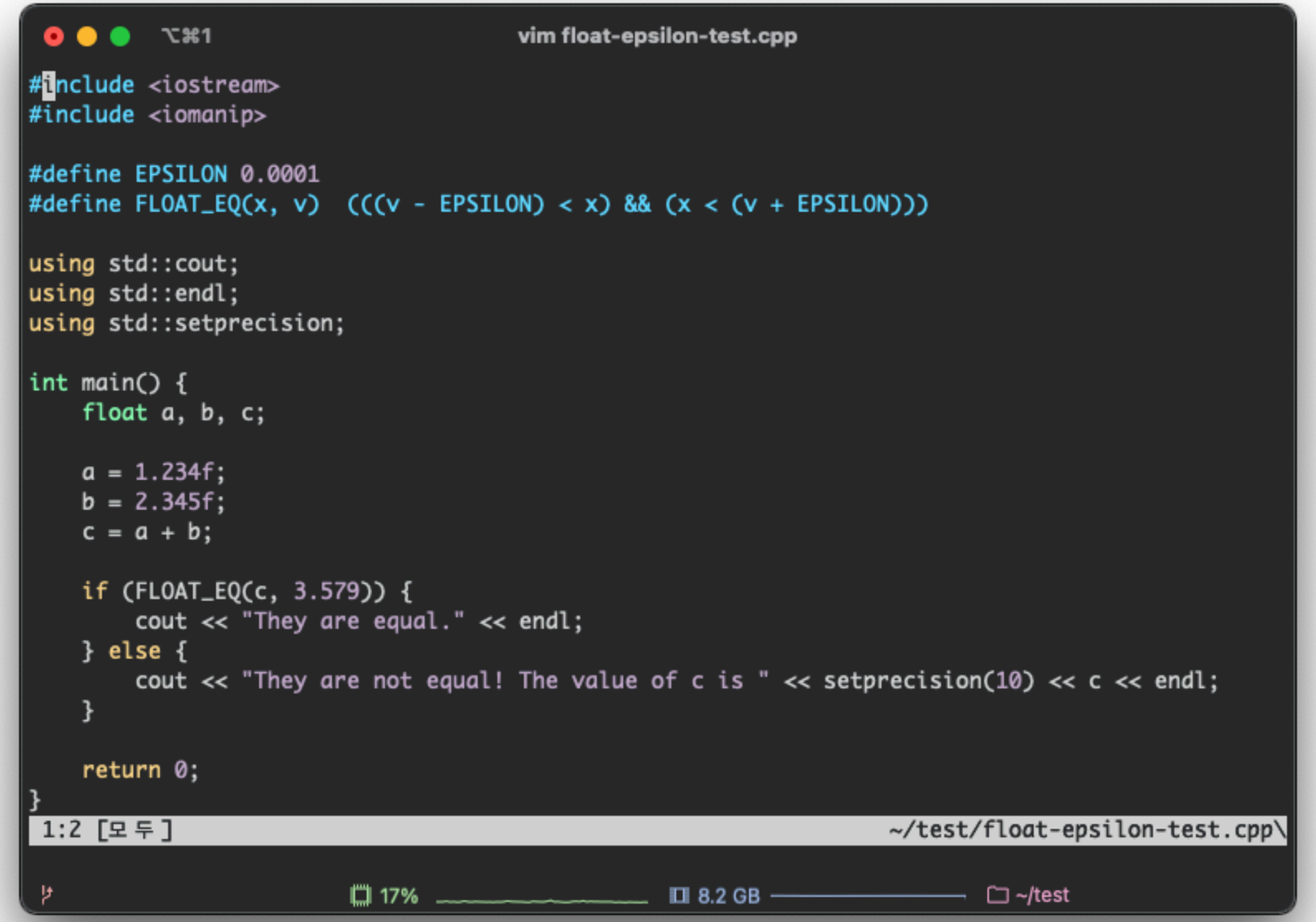
Basic Floating-Point Test



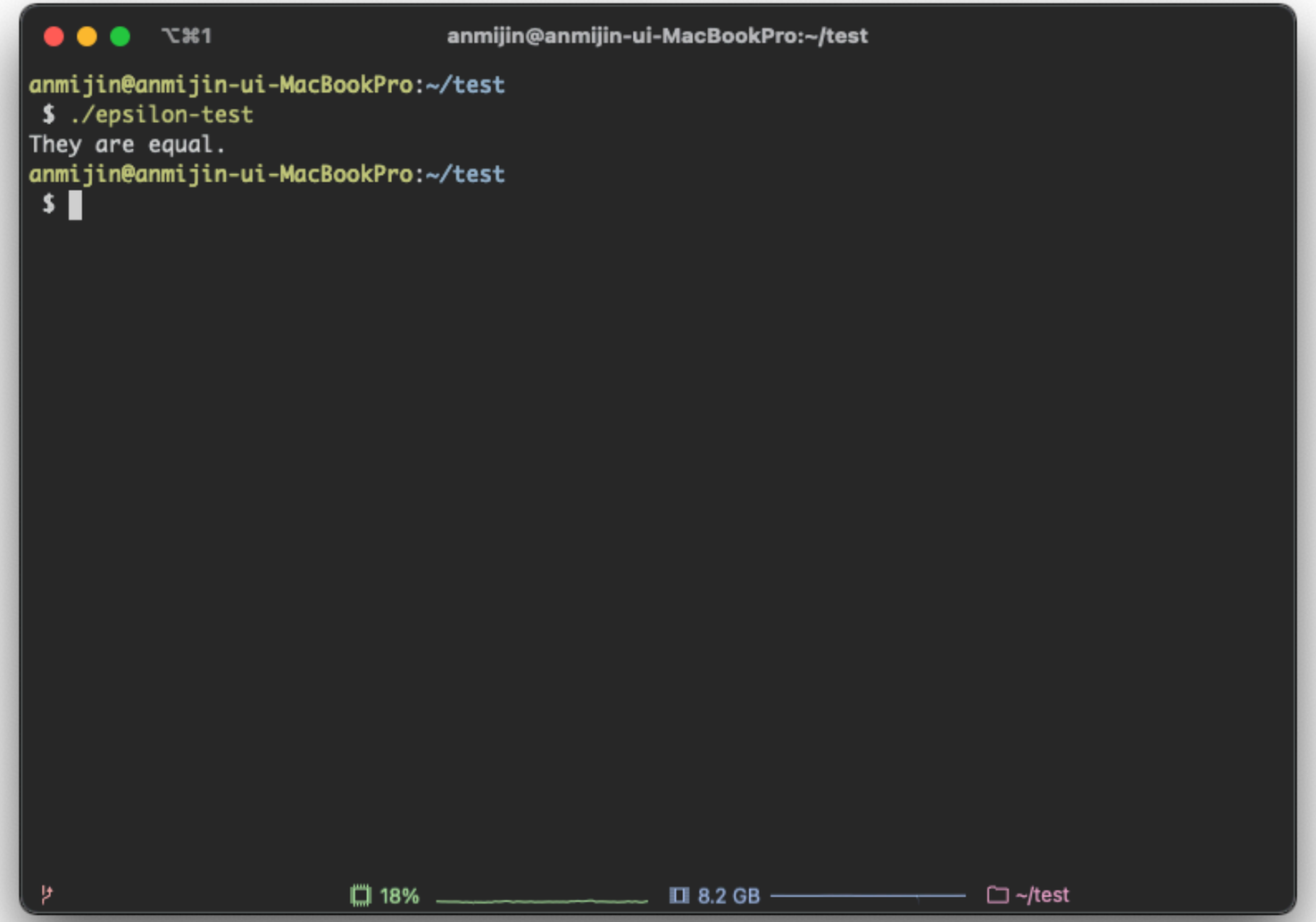
Basic Floating-Point Test: Result



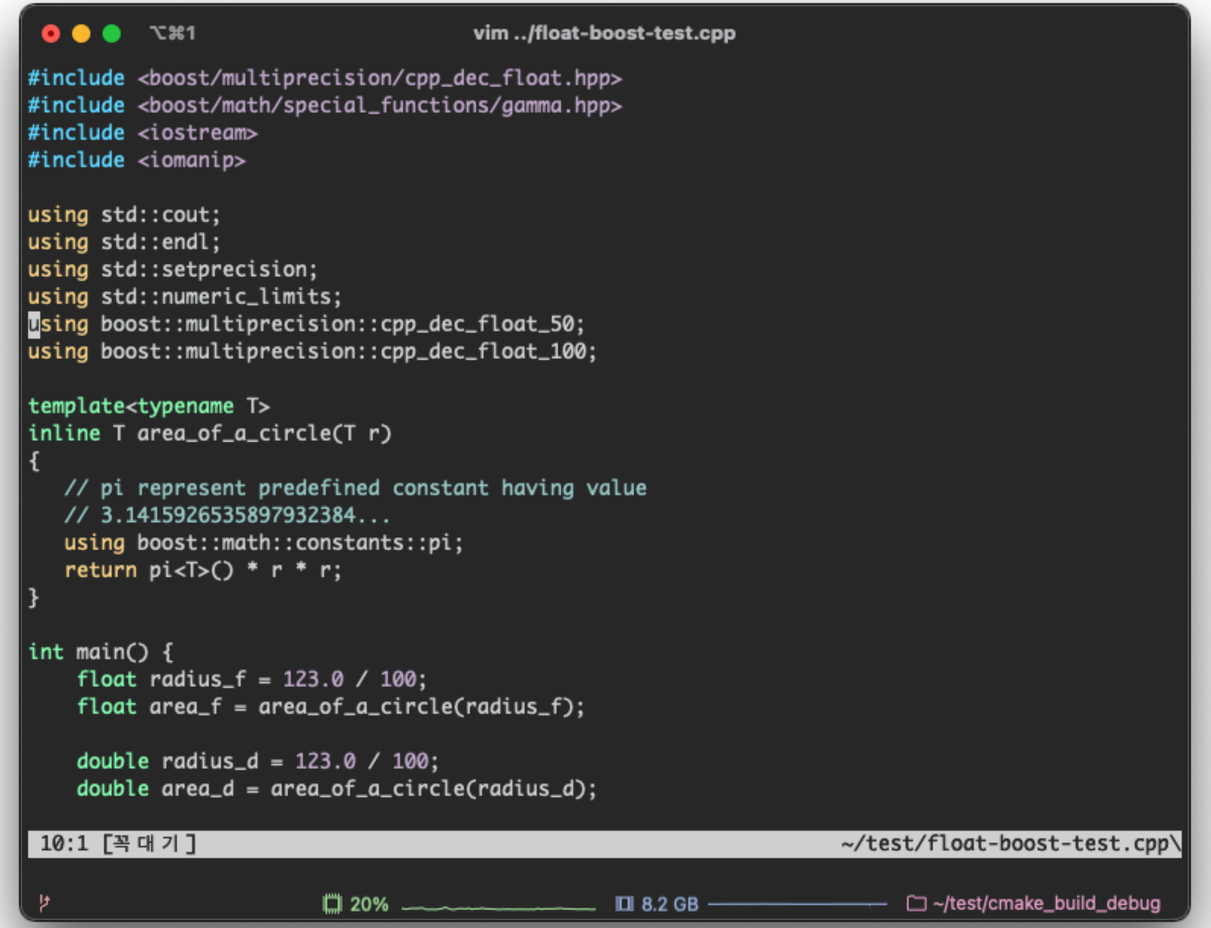
Epsilon Test

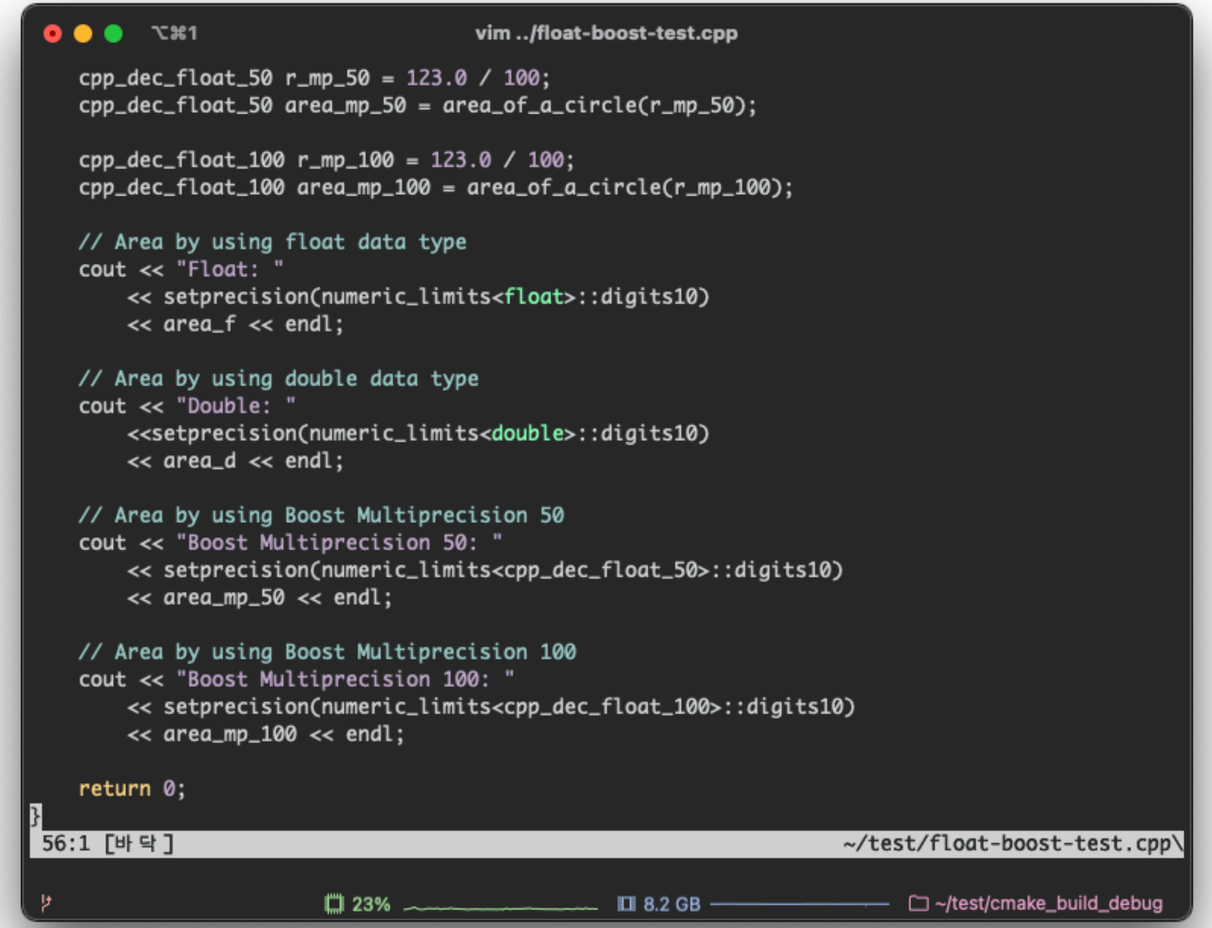


Epsilon Test: Result

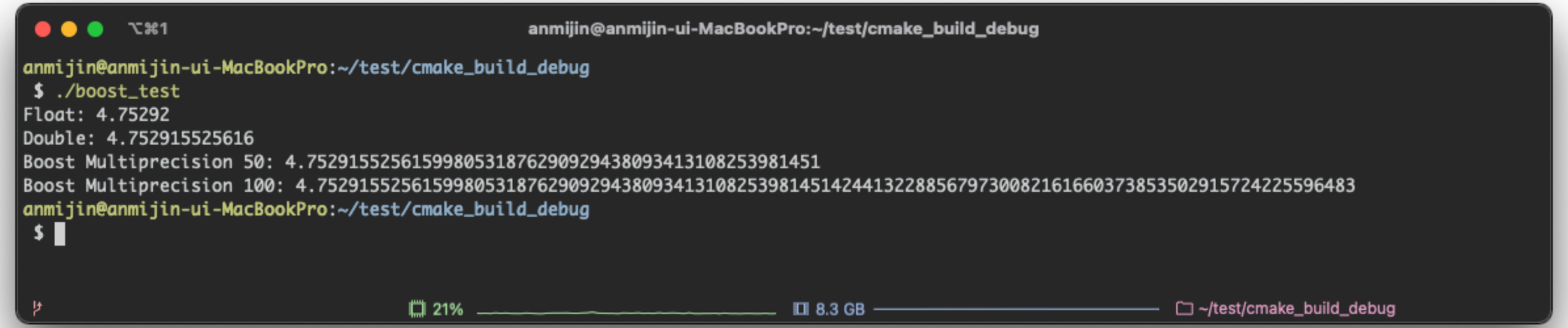


Floating-Point Test With Boost Library





Floating-Point Test With Boost Library: Result



Reference

[1] “Why Floating-Point Numbers May Lose Precision”, Microsoft, https://docs.microsoft.com/en-us/cpp/build/why-floating-point-numbers-may-lose-precision?view=msvc-160

[2] “Advanced C++ with boost library”, GeeksforGeeks, https://www.geeksforgeeks.org/advanced-c-boost-library/

[3] “Floating-point Comparison”, boost, https://www.boost.org/doc/libs/1_67_0/libs/math/doc/html/math_toolkit/float_comparison.html

[4] “IEEE 754”, Wikipedia, https://ko.wikipedia.org/wiki/IEEE_754