

Kaitlyn Ellis

Professor Niu

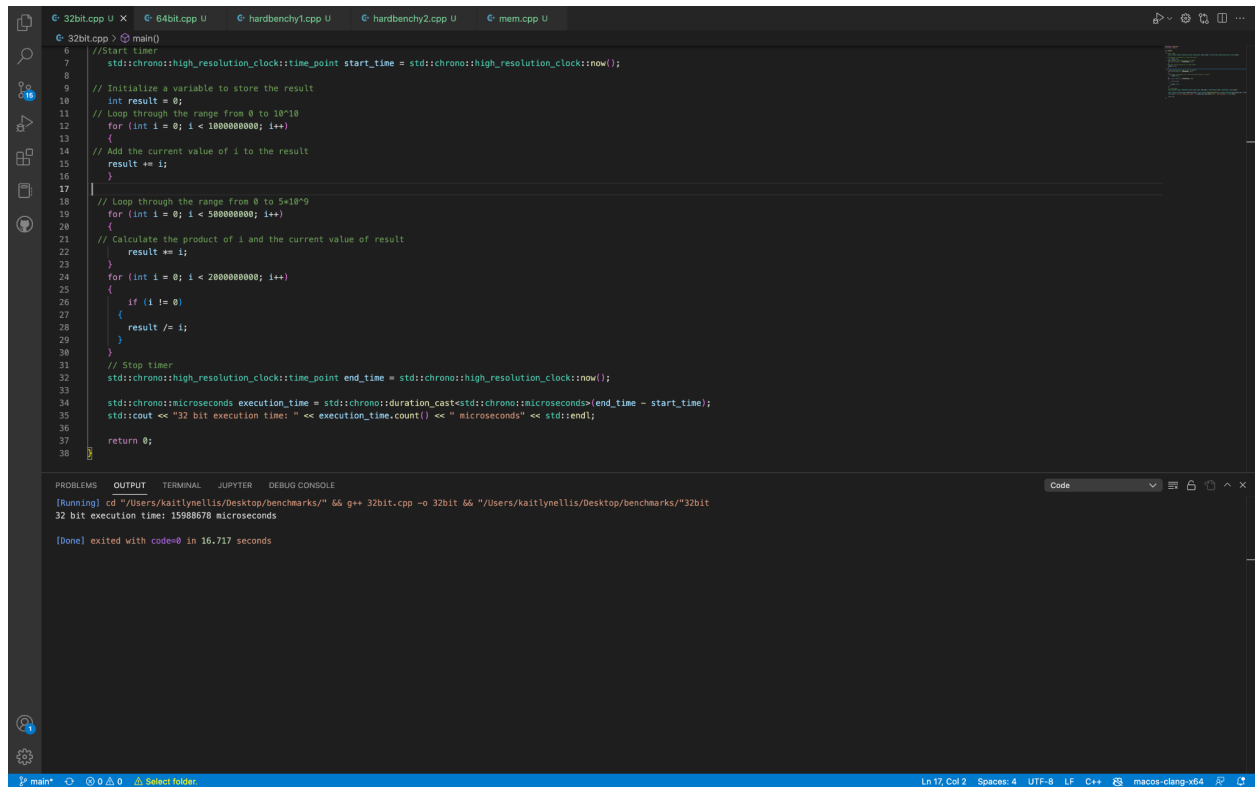
CSCI 202

7 February 2023

Computer Specs:

- CPU: Intel
 - CPU Model: Coffee Lake Core i7
 - # of Cores on CPU: 6
 - CPU Clock Rate in GHz: 2.6
- Memory
 - Amount in GB: 16
 - Speed of memory: 2667 MHz DDR4
- Type of Hard Drive: SSD
 - Capacity of Hard Drive: 512 GB
 - Sequential Write Speed: 18886.2 MB/s
 - Sequential Read Speed: 2820.4 MB/s
 - Random Write Speed: 97.1 MB/s
 - Random Read Speed: 68.2 MB/s

32 bit integer operation benchmark:



The image shows a code editor with a C++ file named `32bit.cpp`. The code implements a benchmark for 32-bit integer operations. It starts by including `<chrono>` and `<iostream>`. The `main` function begins with a timer start, initializes a `result` variable to 0, and then performs three loops: a simple addition loop (0 to 10^8), a multiplication loop (0 to 5*10^9), and a division loop (0 to 2*10^9). After the loops, it stops the timer and prints the execution time in microseconds. The terminal output shows the program running successfully and exiting with a code of 0 in 16.717 seconds.

```
6 //Start timer
7 std::chrono::high_resolution_clock::time_point start_time = std::chrono::high_resolution_clock::now();
8
9 // Initialize a variable to store the result
10 int result = 0;
11 // Loop through the range from 0 to 10^8
12 for (int i = 0; i < 100000000; i++)
13 {
14     // Add the current value of i to the result
15     result += i;
16 }
17
18 // Loop through the range from 0 to 5*10^9
19 for (int i = 0; i < 500000000; i++)
20 {
21     // Calculate the product of i and the current value of result
22     result *= i;
23 }
24 for (int i = 0; i < 2000000000; i++)
25 {
26     if (i != 0)
27     {
28         result /= i;
29     }
30 }
31 // Stop timer
32 std::chrono::high_resolution_clock::time_point end_time = std::chrono::high_resolution_clock::now();
33
34 std::chrono::microseconds execution_time = std::chrono::duration_cast<std::chrono::microseconds>(end_time - start_time);
35 std::cout << "32 bit execution time: " << execution_time.count() << " microseconds" << std::endl;
36
37 return 0;
38 }
```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

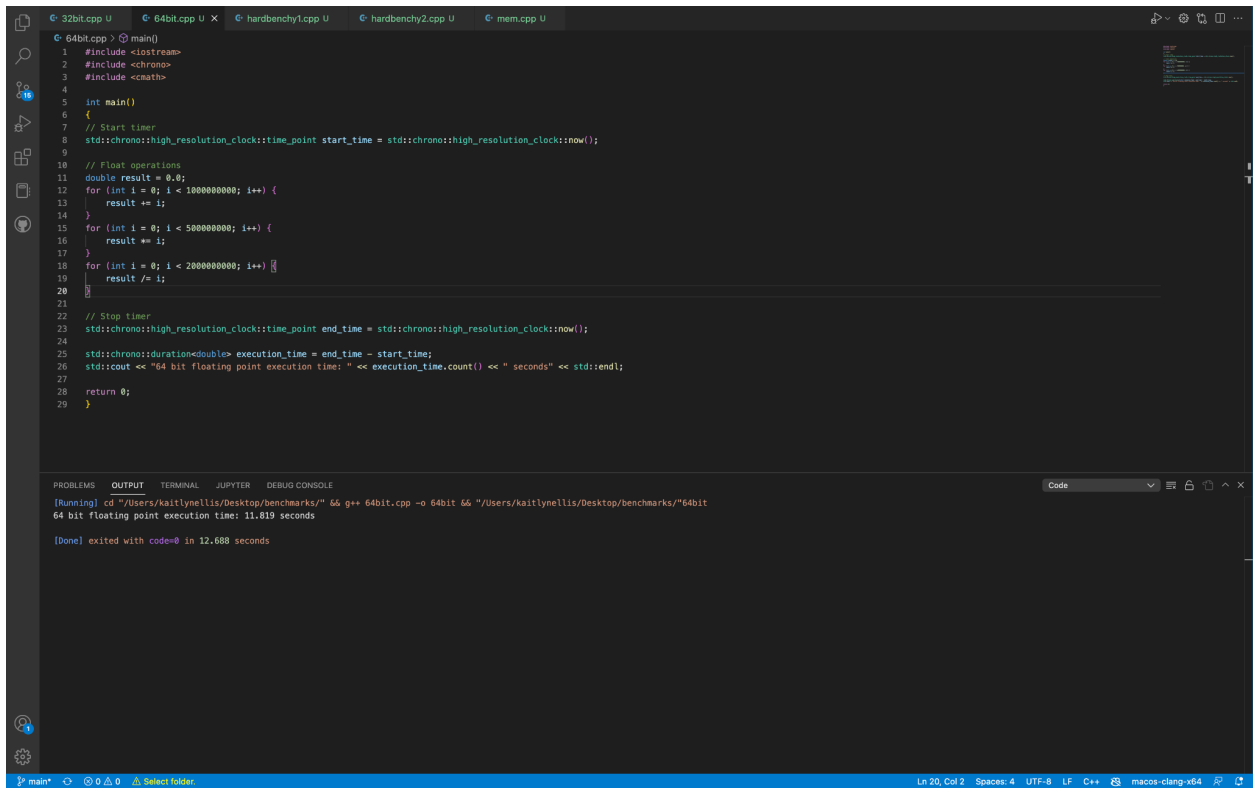
[Running] cd "/Users/kaitlynellis/Desktop/benchmarks/" && g++ 32bit.cpp -o 32bit && "/Users/kaitlynellis/Desktop/benchmarks/"32bit

32 bit execution time: 15988678 microseconds

[Done] exited with code=0 in 16.717 seconds

Ln 17, Col 2 Spaces: 4 UTF-8 LF C++ macos-clang-v64

64 bit Floating point operation benchmark



The image shows a code editor with a C++ file named `64bit.cpp`. The code is a benchmark for 64-bit floating point operations. It includes `<iostream>`, `<chrono>`, and `<cmath>`. The `main` function starts a timer, performs three loops of floating point operations (addition, multiplication, and division), and then stops the timer. The execution time is printed to the console.

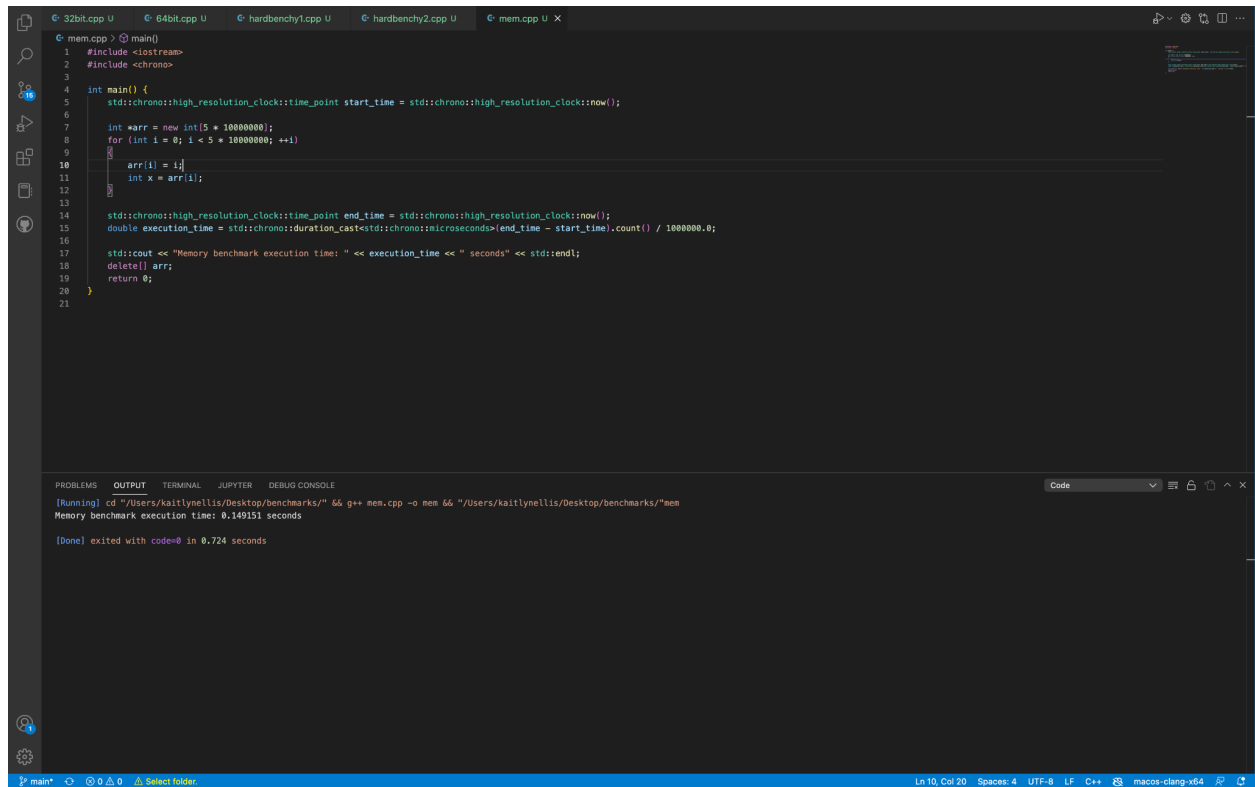
```
1 #include <iostream>
2 #include <chrono>
3 #include <cmath>
4
5 int main()
6 {
7     // Start timer
8     std::chrono::high_resolution_clock::time_point start_time = std::chrono::high_resolution_clock::now();
9
10    // Float operations
11    double result = 0.0;
12    for (int i = 0; i < 1000000000; i++) {
13        result += i;
14    }
15    for (int i = 0; i < 500000000; i++) {
16        result *= i;
17    }
18    for (int i = 0; i < 200000000; i++) {
19        result /= i;
20    }
21
22    // Stop timer
23    std::chrono::high_resolution_clock::time_point end_time = std::chrono::high_resolution_clock::now();
24
25    std::chrono::duration<double> execution_time = end_time - start_time;
26    std::cout << "64 bit floating point execution time: " << execution_time.count() << " seconds" << std::endl;
27
28    return 0;
29 }
```

The output of the program is shown in the terminal window:

```
[Running] cd "/Users/kaitlynellis/Desktop/benchmarks/" && g++ 64bit.cpp -o 64bit && "/Users/kaitlynellis/Desktop/benchmarks/"64bit
64 bit floating point execution time: 11.819 seconds

[Done] exited with code=0 in 12.688 seconds
```

Memory Benchmark:



The screenshot displays a Visual Studio Code editor with a C++ file named `mem.cpp` open. The code implements a memory benchmark that allocates a large array, fills it with sequential integers, and measures the execution time. The output panel shows the program running successfully and reporting an execution time of approximately 0.149 seconds.

```
1 #include <iostream>
2 #include <chrono>
3
4 int main() {
5     std::chrono::high_resolution_clock::time_point start_time = std::chrono::high_resolution_clock::now();
6
7     int *arr = new int[5 * 10000000];
8     for (int i = 0; i < 5 * 10000000; ++i)
9     {
10         arr[i] = i;
11         int x = arr[i];
12     }
13
14     std::chrono::high_resolution_clock::time_point end_time = std::chrono::high_resolution_clock::now();
15     double execution_time = std::chrono::duration_cast<std::chrono::microseconds>(end_time - start_time).count() / 1000000.0;
16
17     std::cout << "Memory benchmark execution time: " << execution_time << " seconds" << std::endl;
18     delete[] arr;
19     return 0;
20 }
21
```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

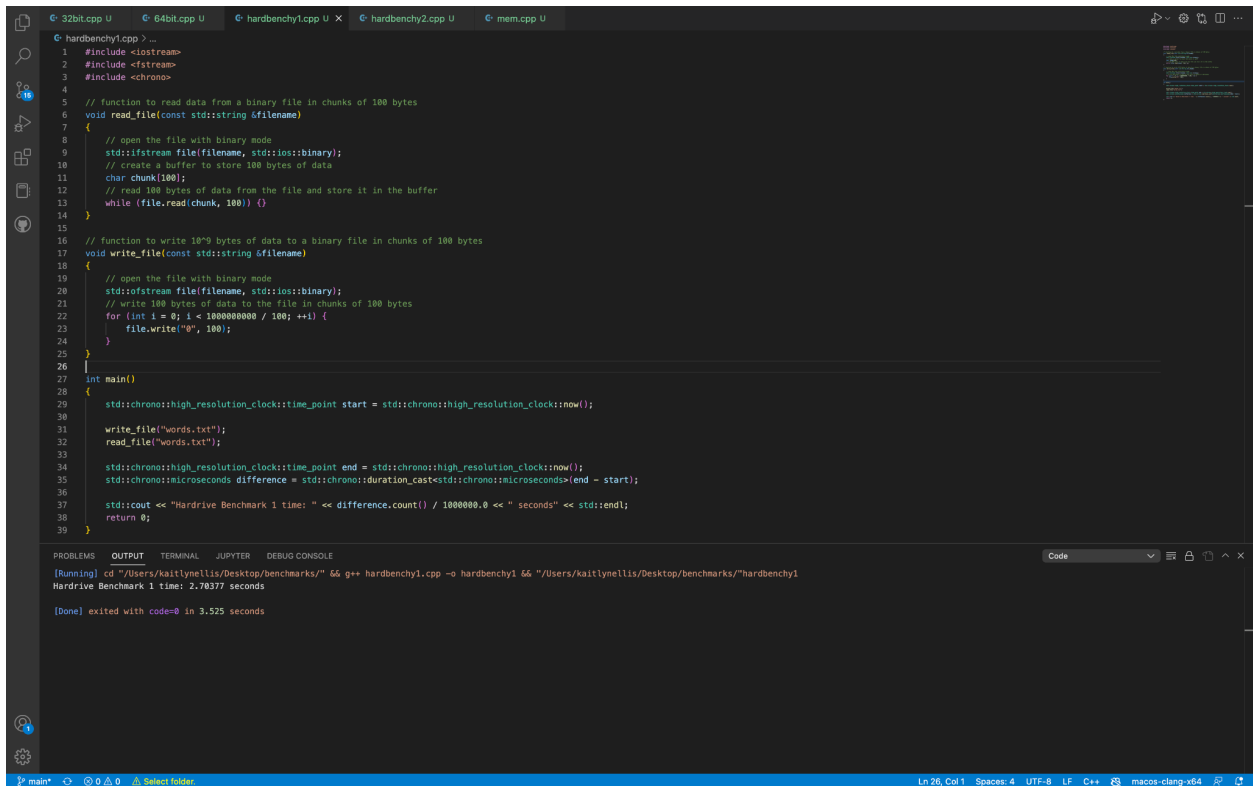
[Running] cd "/Users/kaitlynellis/Desktop/benchmarks/" && g++ mem.cpp -o mem && "/Users/kaitlynellis/Desktop/benchmarks/mem

Memory benchmark execution time: 0.149151 seconds

[Done] exited with code=0 in 0.724 seconds

Ln 10, Col 20 Spaces: 4 UTF-8 LF C++ macOS-clang-v64

Hardrive benchmark #1



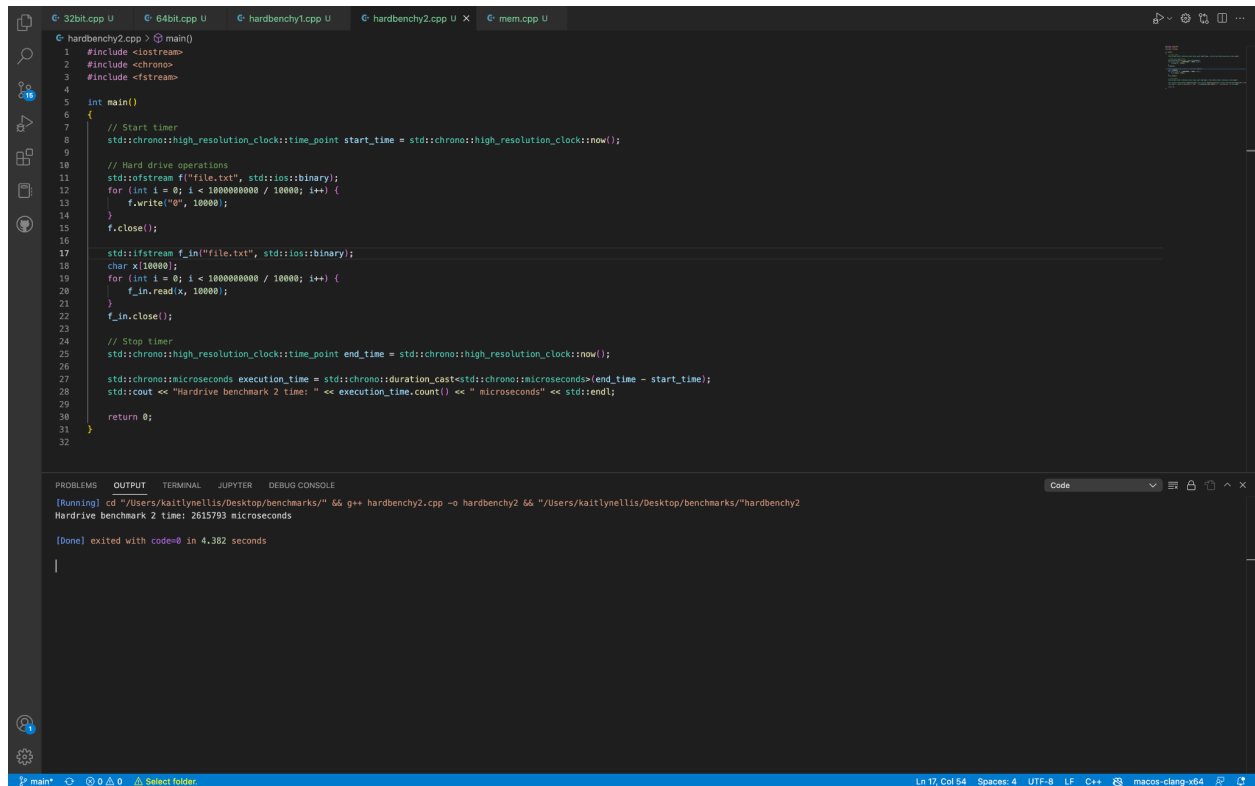
The image shows a C++ IDE with a dark theme. The main editor window displays a C++ program named `hardbench1.cpp`. The program includes `<iostream>`, `<fstream>`, and `<chrono>`. It defines two functions: `read_file` and `write_file`, both taking a filename as a parameter. `read_file` opens a file in binary mode, creates a 100-byte buffer, and reads data in chunks of 100 bytes. `write_file` opens a file in binary mode and writes 100 bytes of data in chunks of 100 bytes. The `main` function uses `std::chrono::high_resolution_clock` to measure the time taken to write and read a file named `words.txt`. It calculates the difference in time and prints the result in seconds.

```
1 #include <iostream>
2 #include <fstream>
3 #include <chrono>
4
5 // function to read data from a binary file in chunks of 100 bytes
6 void read_file(const std::string &filename)
7 {
8     // open the file with binary mode
9     std::ifstream file(filename, std::ios::binary);
10    // create a buffer to store 100 bytes of data
11    char chunk[100];
12    // read 100 bytes of data from the file and store it in the buffer
13    while (file.read(chunk, 100)) {}
14 }
15
16 // function to write 10^9 bytes of data to a binary file in chunks of 100 bytes
17 void write_file(const std::string &filename)
18 {
19     // open the file with binary mode
20     std::ofstream file(filename, std::ios::binary);
21     // write 100 bytes of data to the file in chunks of 100 bytes
22     for (int i = 0; i < 1000000000 / 100; ++i) {
23         file.write("0", 100);
24     }
25 }
26
27 int main()
28 {
29     std::chrono::high_resolution_clock::time_point start = std::chrono::high_resolution_clock::now();
30
31     write_file("words.txt");
32     read_file("words.txt");
33
34     std::chrono::high_resolution_clock::time_point end = std::chrono::high_resolution_clock::now();
35     std::chrono::microseconds difference = std::chrono::duration_cast<std::chrono::microseconds>(end - start);
36
37     std::cout << "Hardrive Benchmark 1 time: " << difference.count() / 1000000.0 << " seconds" << std::endl;
38     return 0;
39 }
```

The terminal window shows the command to compile and run the program: `cd "/Users/kaitlynellis/Desktop/benchmarks/" && g++ hardbench1.cpp -o hardbench1 && "/Users/kaitlynellis/Desktop/benchmarks/"hardbench1`. The output is: `Hardrive Benchmark 1 time: 2.70377 seconds`.

At the bottom of the IDE, the status bar shows: `Ln 26, Col 1 Spaces: 4 UTF-8 LF C++ macos-clang-x64`.

Hardrive Benchmark #2:



The screenshot shows a C++ IDE with a file named `hardbench2.cpp` open. The code is a benchmark for hard drive operations. It starts by including `<iostream>`, `<chrono>`, and `<fstream>`. In the `main` function, it starts a timer, performs hard drive operations (writing and reading 100,000 bytes), stops the timer, and prints the execution time in microseconds. The output window shows the command used to run the program and the resulting execution time of 2615793 microseconds.

```
1 #include <iostream>
2 #include <chrono>
3 #include <fstream>
4
5 int main()
6 {
7     // Start timer
8     std::chrono::high_resolution_clock::time_point start_time = std::chrono::high_resolution_clock::now();
9
10    // Hard drive operations
11    std::ofstream f("file.txt", std::ios::binary);
12    for (int i = 0; i < 1000000000 / 10000; i++) {
13        f.write("0", 10000);
14    }
15    f.close();
16
17    std::ifstream f_in("file.txt", std::ios::binary);
18    char x[10000];
19    for (int i = 0; i < 1000000000 / 10000; i++) {
20        f_in.read(x, 10000);
21    }
22    f_in.close();
23
24    // Stop timer
25    std::chrono::high_resolution_clock::time_point end_time = std::chrono::high_resolution_clock::now();
26
27    std::chrono::microseconds execution_time = std::chrono::duration_cast<std::chrono::microseconds>(end_time - start_time);
28    std::cout << "Hardrive benchmark 2 time: " << execution_time.count() << " microseconds" << std::endl;
29
30    return 0;
31 }
32
```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

[Running] cd "/Users/kaitlynellis/Desktop/benchmarks/" && g++ hardbench2.cpp -o hardbench2 && "/Users/kaitlynellis/Desktop/benchmarks/hardbench2"

Hardrive benchmark 2 time: 2615793 microseconds

[Done] exited with code=0 in 4.382 seconds

Ln 17, Col 54 Spaces: 4 UTF-8 LF C++ macos-clang-x64

Type of Benchmark	Result
32 bit	15988678 microseconds
64 bit	11.819 seconds
Memory	0.149151 seconds
HD #1	2.70377 seconds
HD #2	2615793 microseconds

Geometric Mean: 2.8834904362823 seconds