How to use GoogleTest

How to Install GoogleTest

- Install CMake: https://cmake.org/download/
    - After installation, add CMake to the path if you are using Mac:
    - sudo "/Applications/CMake.app/Contents/bin/cmake-gui" --install=/usr/local/bin

How to Install GoogleTest on Mac [1]

- To recommend installing it on your own computer, Since we don't have sudo permissions on the lab computers
- If you use Windows, you might need to refer to other Internet resources

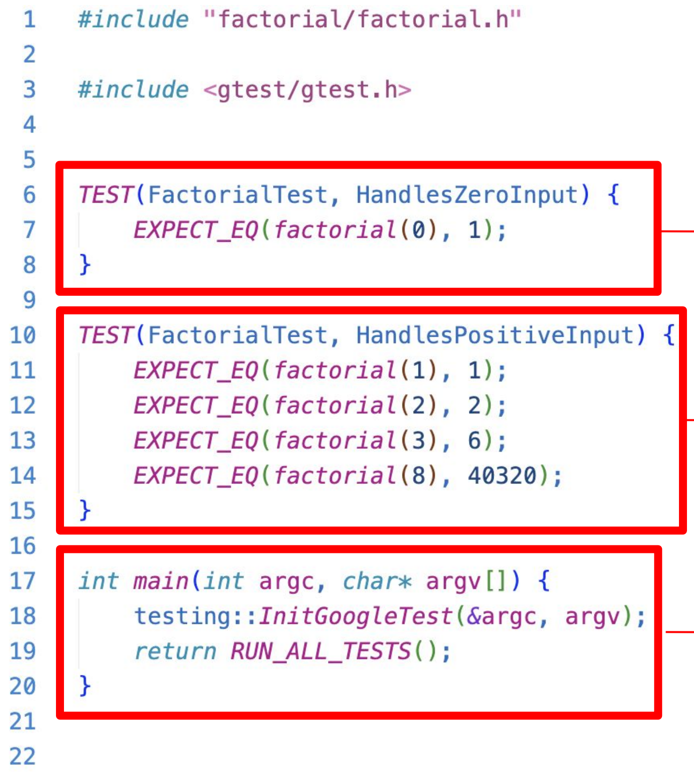
```
git clone https://github.com/google/googletest.git -b v1.16.0
cd googletest/googletest  # Main directory of the cloned repository.
mkdir build              # Create a directory to hold the build output.
cmake ..             # Generate native build scripts for GoogleTest.
make
sudo make install    # Install in /usr/local/ by default
```

[1] https://github.com/google/googletest/blob/main/googletest/README.md

How to use GoogleTest

- After you successfully install the GoogleTest
- Download the factorial.zip provided in Blackboard under week 8D
    - Provided you with:
    - Header file
    - Function file - contain the functions we want to test
    - Main function file - call function file and use GoogleTest to test it

In factorial/main.cpp



```cpp
#include "factorial/factorial.h"

#include <gtest/gtest.h>


TEST(FactorialTest, HandlesZeroInput) {
    EXPECT_EQ(factorial(0), 1);
}

TEST(FactorialTest, HandlesPositiveInput) {
    EXPECT_EQ(factorial(1), 1);
    EXPECT_EQ(factorial(2), 2);
    EXPECT_EQ(factorial(3), 6);
    EXPECT_EQ(factorial(8), 40320);
}

int main(int argc, char* argv[]) {
    testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}

```

One test that contains one case

One test that contains multiple cases

Main function that initializes the Google Test framework and run all the tests declared with the `TEST(...)`

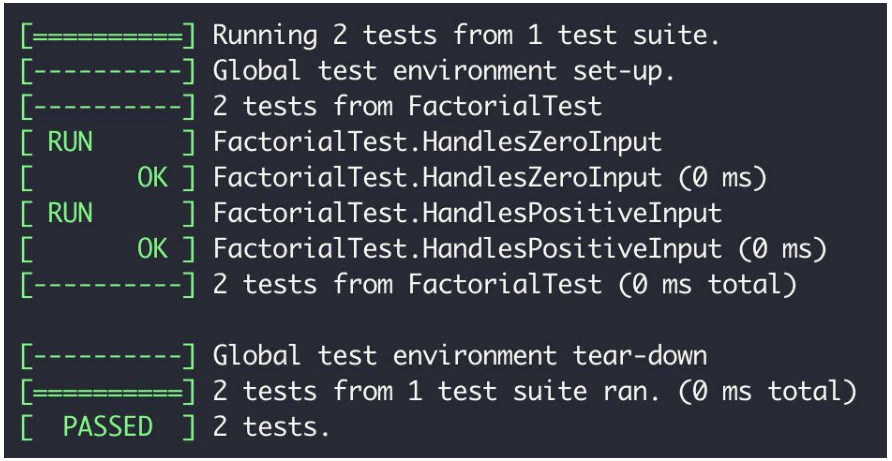
Run GoogleTest

Go into the factorial folder, then compile those files with:

```
g++ -std=c++17 main.cpp factorial_correct.cpp -lgtest -lgtest_main -pthread -o factorial_correct
```

Run GoogleTest if the code pass all the tests:

```
./factorial_correct
```



```
[==========] Running 2 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 2 tests from FactorialTest
[ RUN      ] FactorialTest.HandlesZeroInput
[       OK ] FactorialTest.HandlesZeroInput (0 ms)
[ RUN      ] FactorialTest.HandlesPositiveInput
[       OK ] FactorialTest.HandlesPositiveInput (0 ms)
[----------] 2 tests from FactorialTest (0 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 1 test suite ran. (0 ms total)
[  PASSED  ] 2 tests.
```

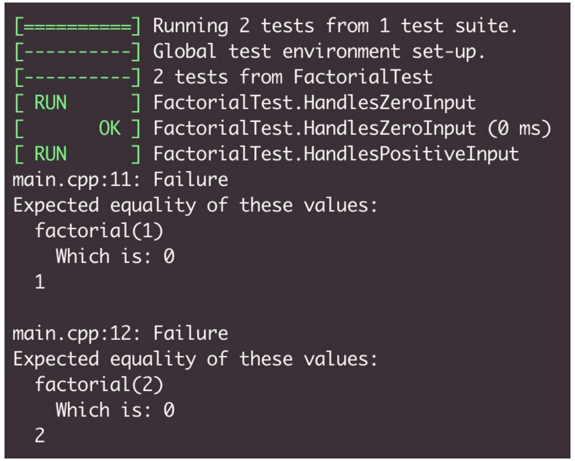
Run GoogleTest

Go into the factorial folder, then compile those files with:

```
g++ -std=c++17 main.cpp factorial_wrong.cpp -lgtest -lgtest_main -pthread -o factorial_wrong
```

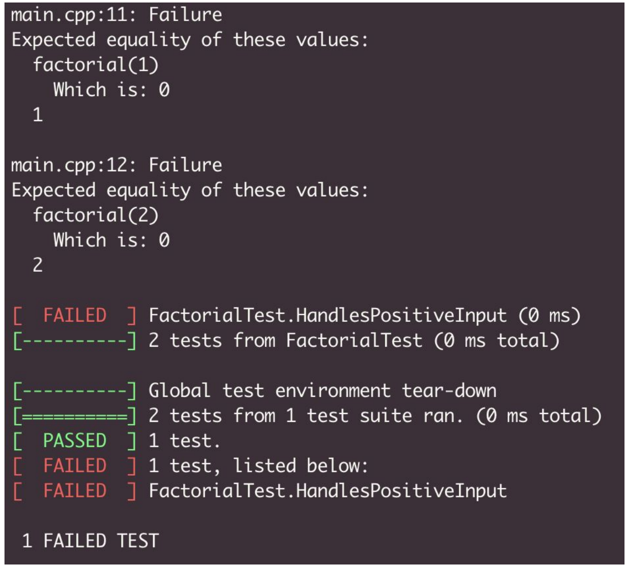
Run GoogleTest if the code didn't pass all the tests:

```
./factorial_wrong
```



```
[==========] Running 2 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 2 tests from FactorialTest
[ RUN      ] FactorialTest.HandlesZeroInput
[       OK ] FactorialTest.HandlesZeroInput (0 ms)
[ RUN      ] FactorialTest.HandlesPositiveInput
main.cpp:11: Failure
Expected equality of these values:
  factorial(1)
    Which is: 0
  1

main.cpp:12: Failure
Expected equality of these values:
  factorial(2)
    Which is: 0
  2
```



```
main.cpp:11: Failure
Expected equality of these values:
  factorial(1)
    Which is: 0
  1

main.cpp:12: Failure
Expected equality of these values:
  factorial(2)
    Which is: 0
  2

[  FAILED  ] FactorialTest.HandlesPositiveInput (0 ms)
[----------] 2 tests from FactorialTest (0 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 1 test suite ran. (0 ms total)
[  PASSED  ] 1 test.
[  FAILED  ] 1 test, listed below:
[  FAILED  ] FactorialTest.HandlesPositiveInput

 1 FAILED TEST
```

Links to Learn More

[1] https://google.github.io/googletest/primer.html