How to install OpenMP in Xcode:

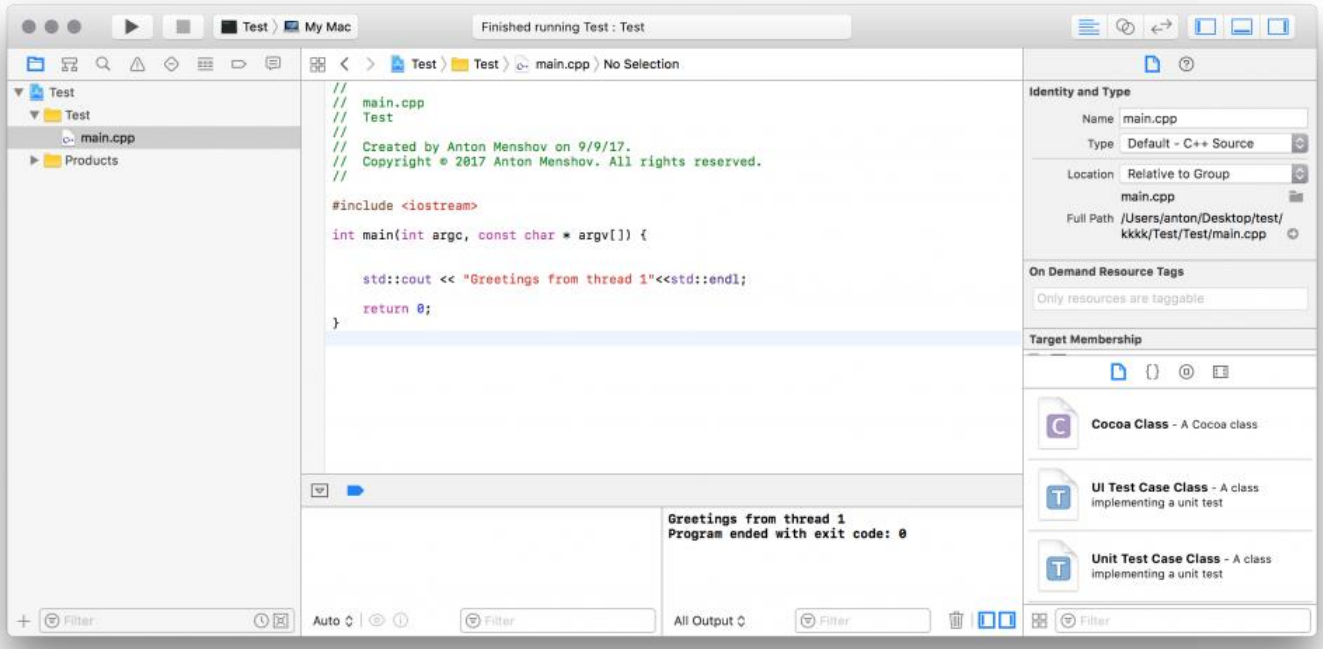
The compiler that is installed on Mac with Xcode does not have all the necessary features to comple OpenMP. I suggest installation of *OpenMP* via *Homebrew*.

1. Install Homebrew if it is not installed on your Mac machine: go to https://brew.sh/
2. Install *libomp* by pasting the following command in your Terminal:

```
brew install libomp
```

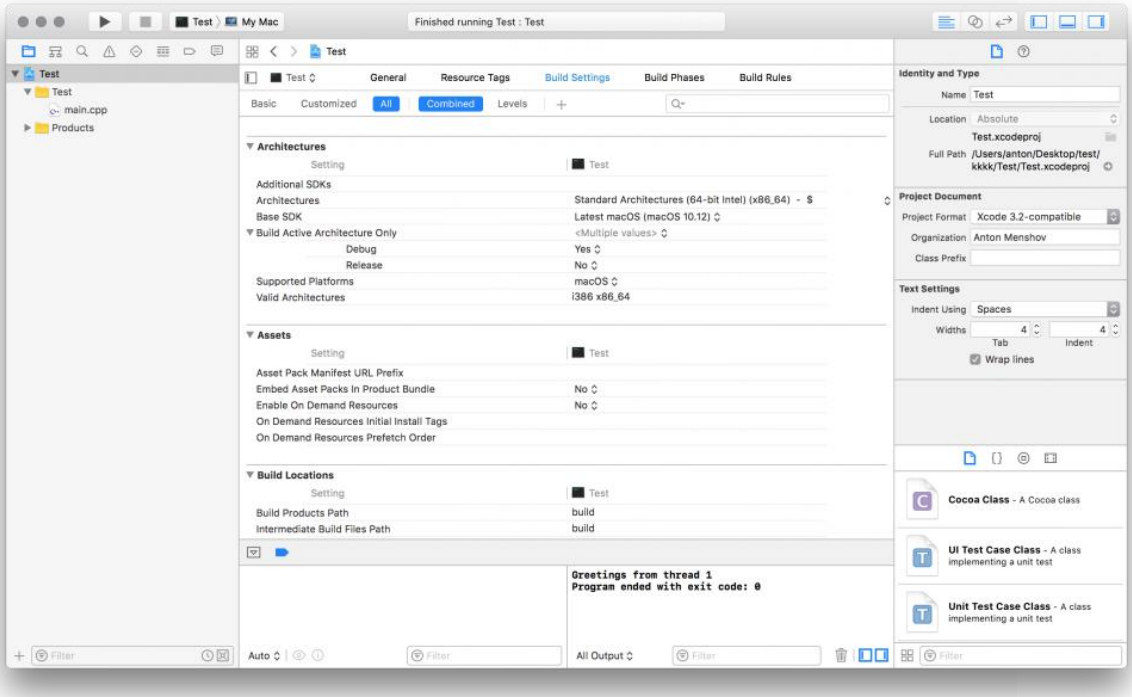
The following steps will be explained based on a “dummy project”; however, it should work for the existing project as well. The following code example is a standard **one-thread** HelloWorld.

```
#include <iostream>
int main(int argc, const char * argv[]) {
    std::cout << "Greetings from thread 1"<<std::endl;
    return 0;
}
```

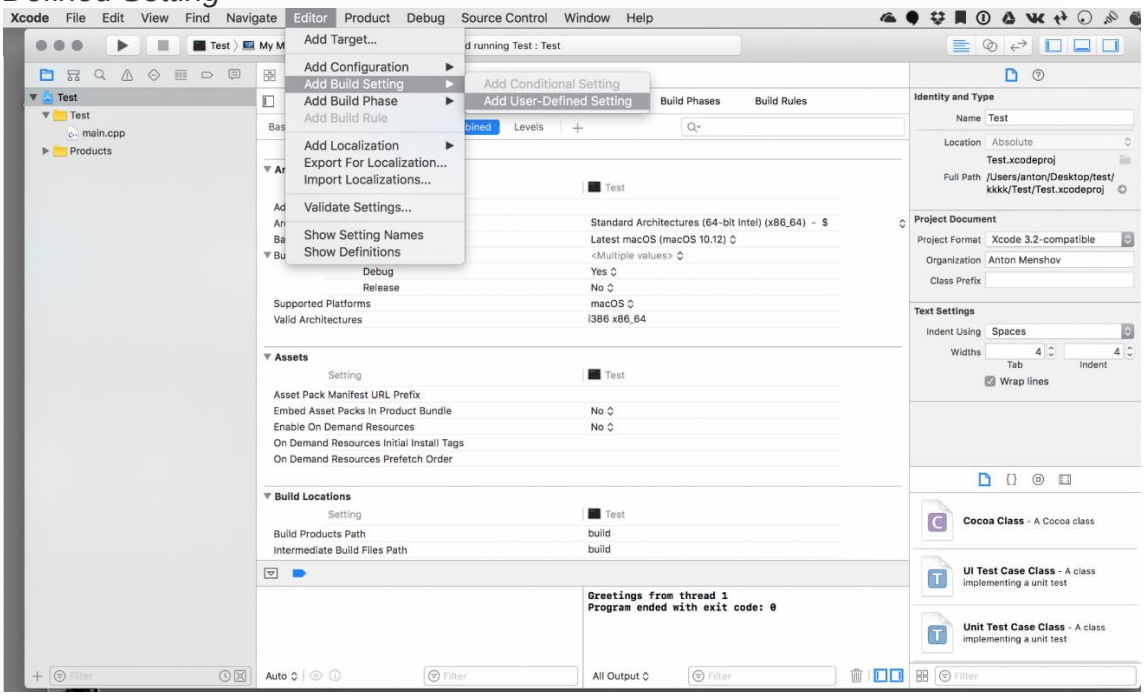


3. Select the project in the *Project Explorer* in the left and select *Build Settings CC* has to be changed accordingly.)

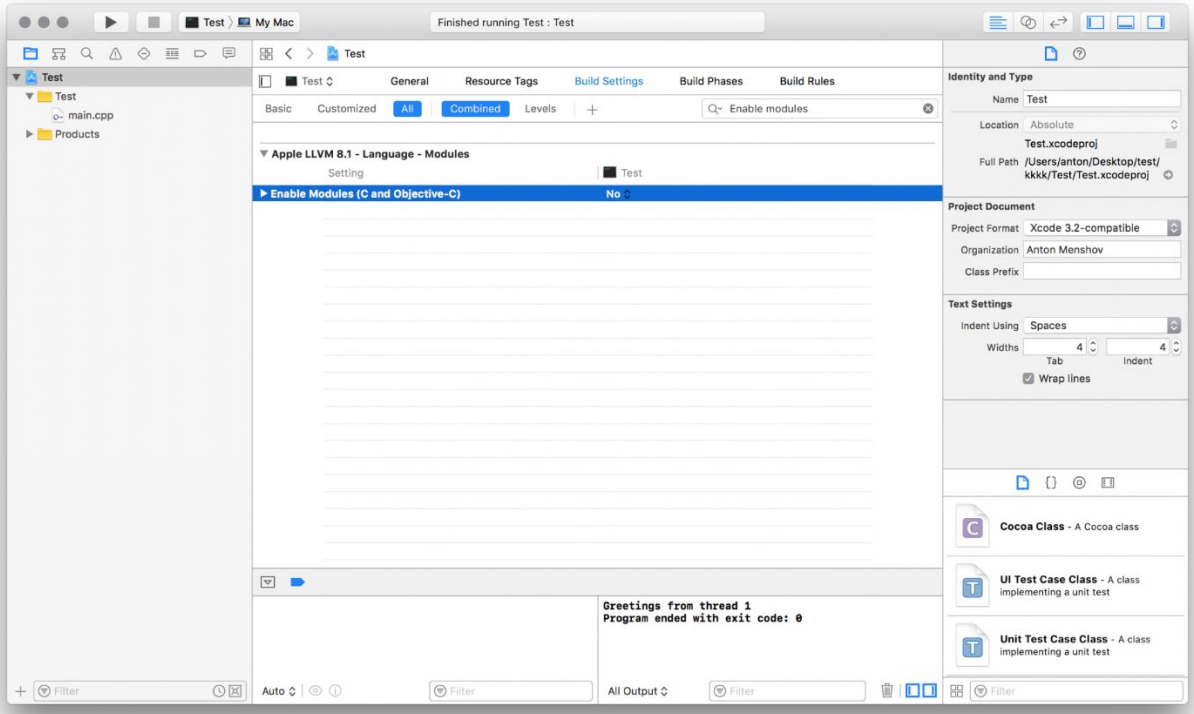
   a.



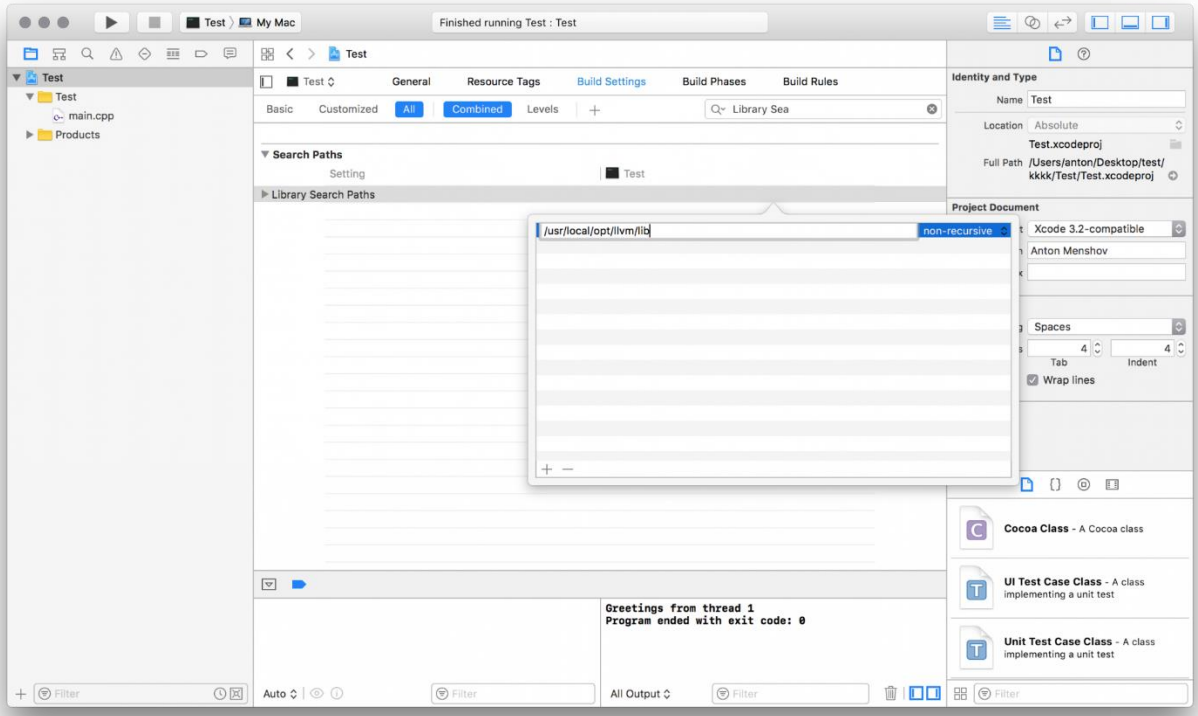
   b. Select from the main Xcode menu*Editor -> Add Build Setting -> Add User-Defined Setting*



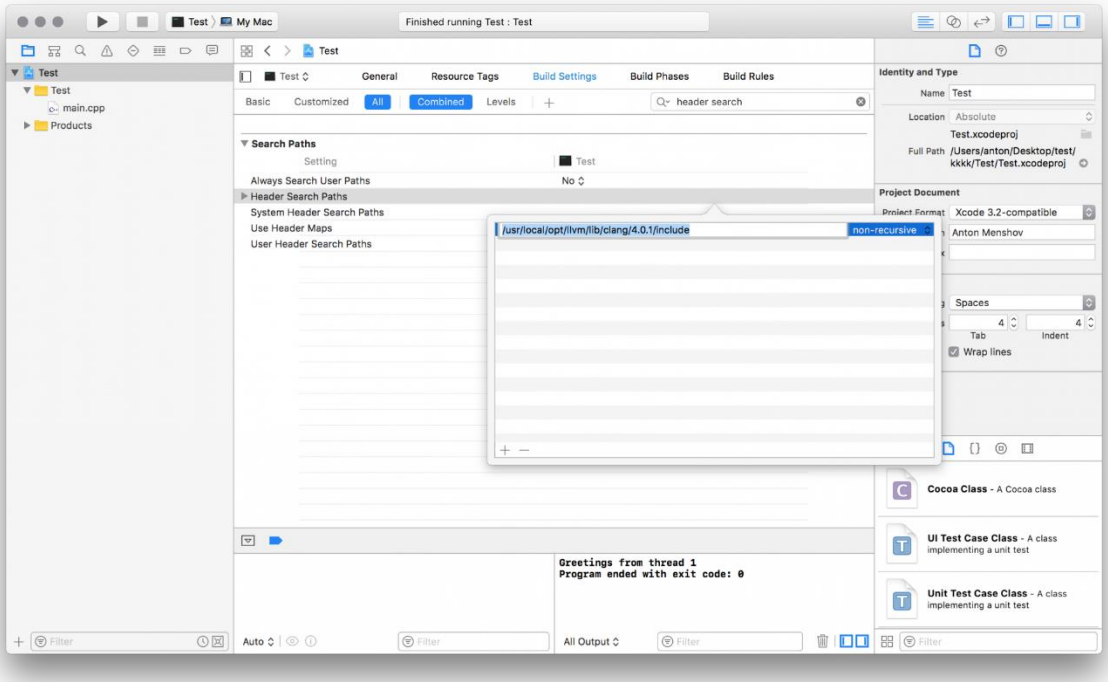
4. Search for the setting *Enable Modules (C and Objective-C)* and switch it from *Yes* to *No*



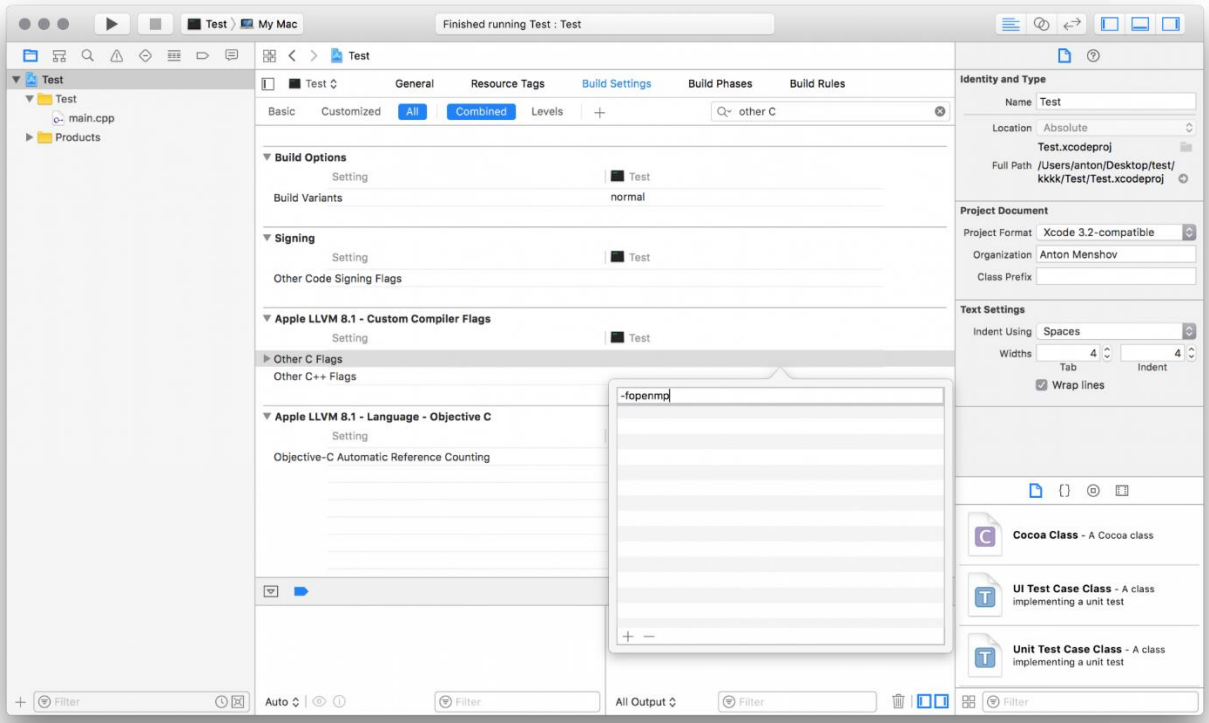
5. Add to the *Library Search Paths* the value */usr/local/opt/libomp/lib* to allow for the precompiled libraries (for example, *libiomp5*) to be dynamically linked to your executable.



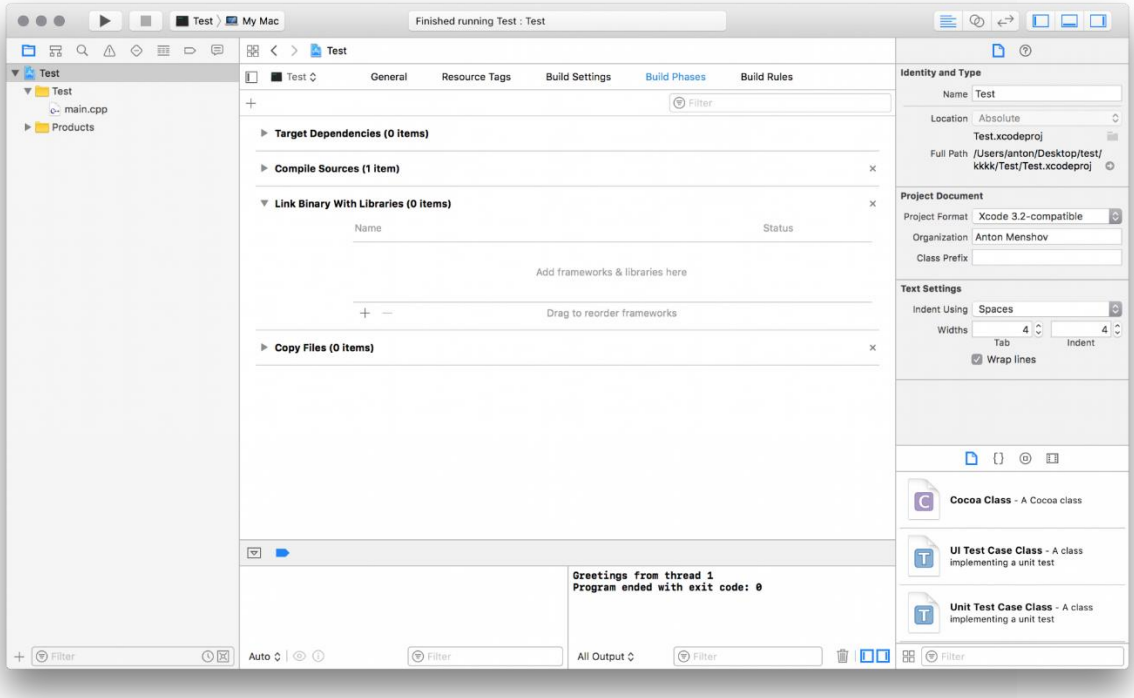
6. Add to the *Header Search Paths* the value */usr/local/opt/libomp/include* to allow inclusion of headers provided by libomp installation (for example, *omp.h*). Note, the installed version of Clang compiler might be different, so the value of the *Header Search Path* should be changed accordingly.



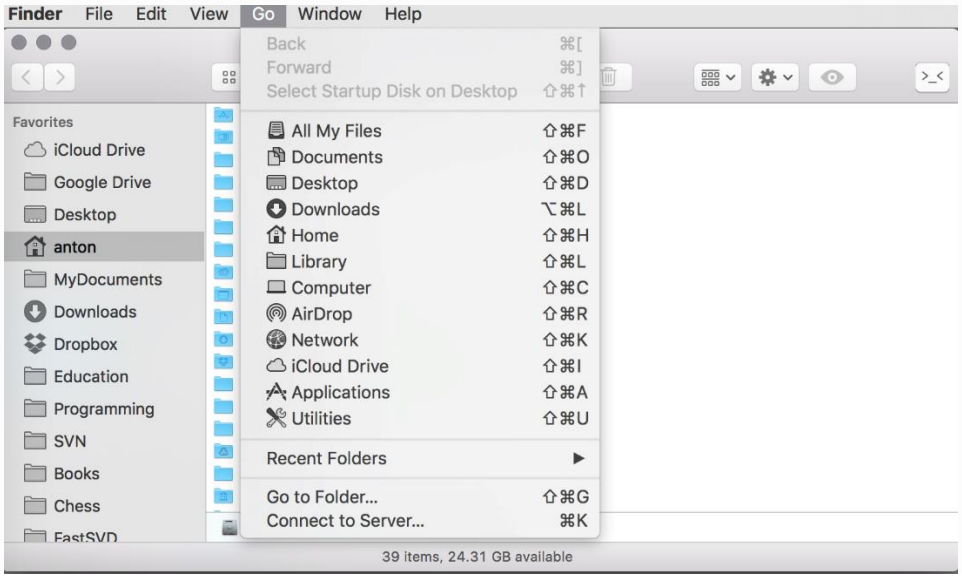
7. Add the *-Xpreprocessor -fopenmp* compilation flag to *Other C Flags*



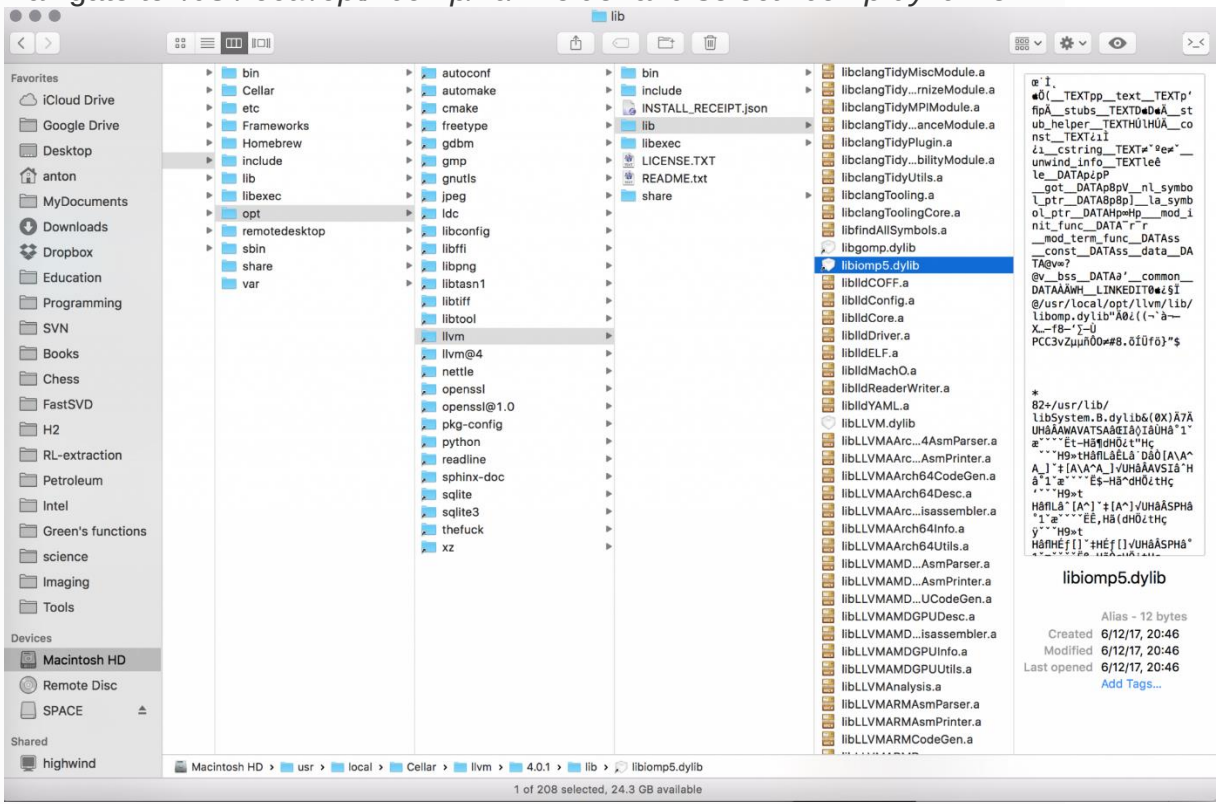
8. Link OpenMP library dynamically to your executable. To do that:
    a. Select *Build Phases -> Link Binary With Libraries* options in the main Xcode window



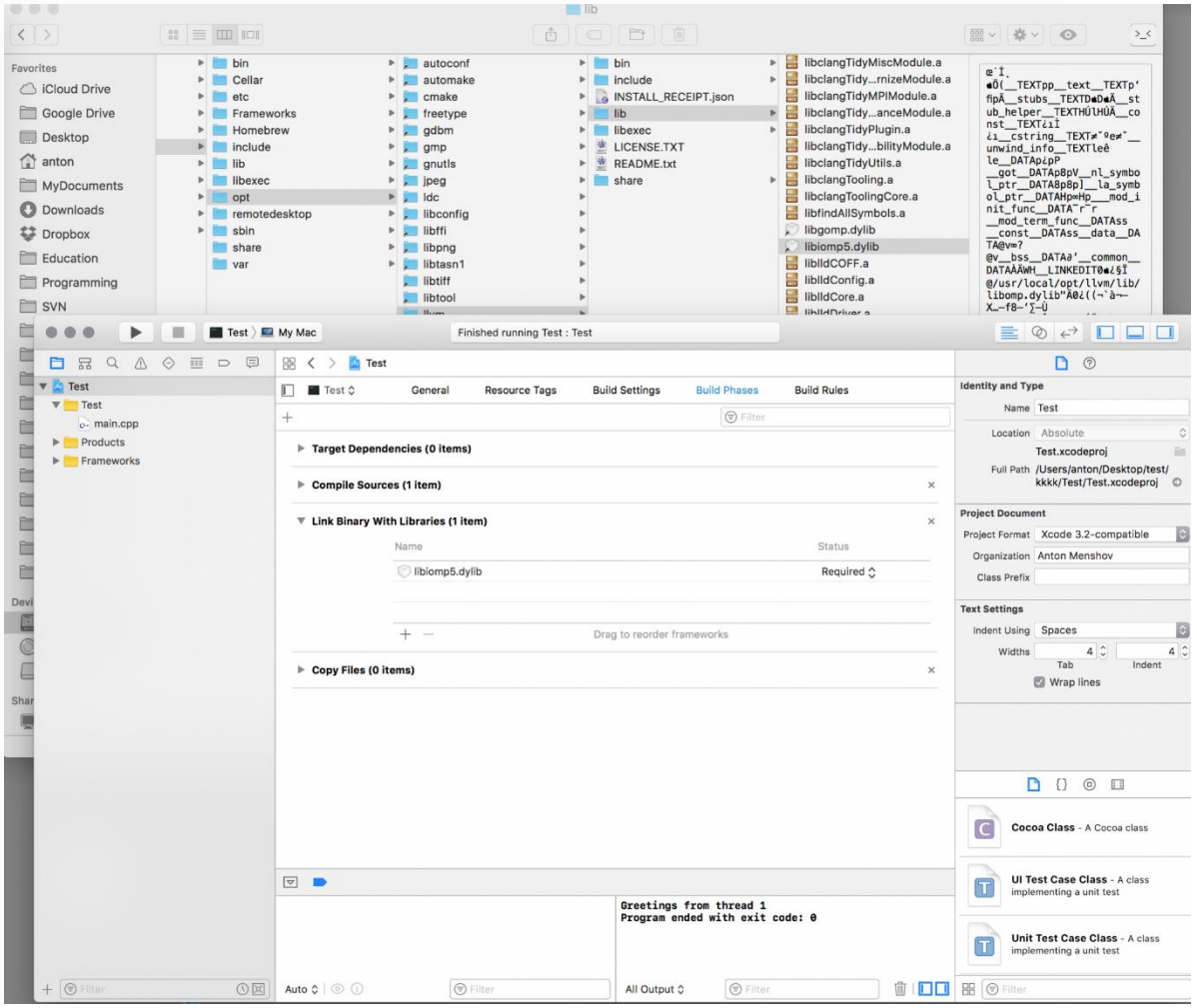
    b. Open a Finder window and select *Go To Folder* option from the main menu.



c. Navigate to */usr/local/opt/libomp/lib/* folder and select *libomp.dylib* file.



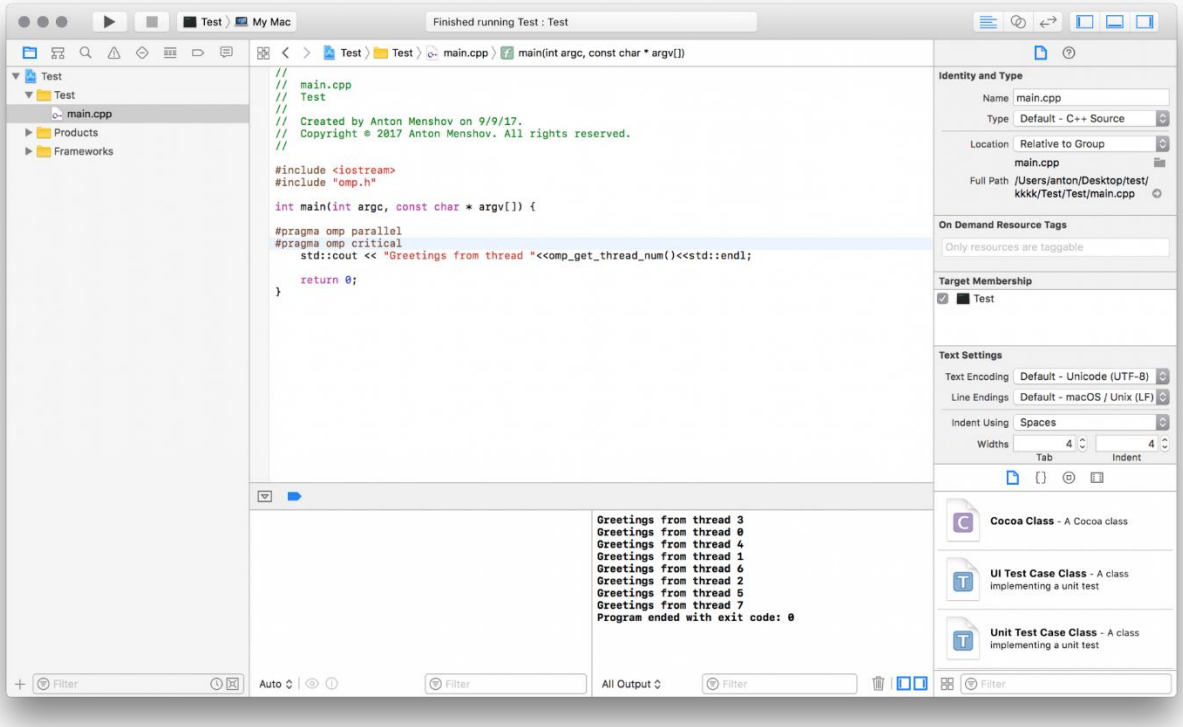
d. Drag it to Xcode main window into the *Add Frameworks and Libraries Here*



Now the code should compile successfully, but already with the compiler.
Slightly editing the code to demonstrate that OpenMP is actually active:

```
1
2   #include <iostream>
3   #include "omp.h"
4   int main(int argc, const char * argv[]) {
5   #pragma omp parallel
6   #pragma omp critical
7       std::cout << "Greetings from thread "<<omp_get_thread_num()<<std::endl;
8       return 0;
    }
```

Now the code greets us from all the threads (*omp critical* section is used so that the output is not “scrambled”)



The number of threads being used is usually controlled via environmental variable that can be edited via Terminal:

```
export OMP_NUM_THREADS=8
```