1. Write a compilation shell script to compile “star_pattern.c” using GCC.

2. For the previous example, use shell variables to represent input filename, output filename, compiler name, and compilation options, and write a more advanced compilation shell script using these variables.

3. Write a shell script to compile “voronoi_1.cpp” into a binary “voronoi_1” using **G++** and dynamic linking. This program uses external dependencies which must be installed first (libcgal, libgmp). Use the provided script install_cgal-4.14.1.sh to install CGAL. Use the `-std=c++11` option.

4. Provide a one-liner command to run the compiled binary executable “voronoi_1” (mind the need to export the directory with CGAL shared library.

5. Write a compilation script (with shell variables) to compile the project “Graph-Executor” using G++. The target executable is `test`, all other files compile to non-executable objects.
   Use these options (search the GCC docs to find the correct flag):
   a. Enable all warnings about questionable constructions
   b. Enable extra warning flags
   c. Compile for `c++14` language standard
   d. Generate debug information
   e. When linking `test`, additionally use `-pthread`

6. Write a Makefile to compile “star_pattern.c” using GCC. Provide an “all” target to build the program an a “clean” target to remove the built binary.

7. For the previous example, write a Makefile with variables to represent compiler name and compilation flags and wildcards to represent input/output filenames.

8. Write a Makefile (with shell variables, wildcards) to compile the project “Graph-Executor” using G++. The target executable is `test`, all other files compile to non-executable objects.
Use the same compilation options as previously (cf Task 5).

9. Use CMake to build the Mesh_2 example in the CGAL library:
    a. Download the latest CGAL from https://github.com/CGAL/cgal/archive/refs/tags/v5.3.tar.gz using wget
    b. Extract the archive contents (hint: use `tar -xzvf <filename>`)
    c. **Follow the instructions** in `<CGAL>/INSTALL.md` to build the code in `<CGAL>/Mesh_2/examples/Mesh_2` directory using CMake and make