Question 1

Suppose you are making a “scientific calculator” that can perform many complex mathematical operations. For this question, let’s assume it can perform ONLY 4 operations: sin, cos, tan, and square root. Now, say you are debugging that program, so you keep constantly changing the code, compiling and running it to see its output. Every time you have to execute this set of instructions:-

```
g++ -c scientific/cos.cpp -o cos.o
g++ -c scientific/sin.cpp -o sin.o
g++ -c scientific/square_root.cpp -o square_root.o
g++ -c scientific/tan.cpp -o tan.o
g++ cos.o sin.o square_root.o tan.o main.cpp -o calculator
./calculator
```

Your friend suggests you to automate this process using “make”. With one command line, you can save time typing this big set of instructions above repeatedly.
If any .cpp file (except main.cpp) is modified, its corresponding .o file should be recompiled. If main.cpp is modified, the executable file “calculator” is recompiled.
.cpp files for sin, cos, tan and square root are present in “scientific/” directory, don’t worry about the “.h” or header files present in that directory.

Make 2 Phony targets, **test,** and **clean**. Whenever you want to **run** your executable, you should type “make test” which in turn should automatically execute “./calculator”. When you do “make clean”, it should remove all the .o files and the executable from the current directory.

Note:-

1) **test** must have the executable as its dependency and **clean** has none.
2) The executable must have dependencies with ALL the .o files.
3) Each .o will have a dependency with its RESPECTIVE .cpp file.
4) As seen in the set of instructions above, don’t forget to use the flag “-c” when compiling to make the .o files and DO NOT use “-c” when compiling for the executable
5) Run “make test” to see the output of the executable.

Side Note:- Here, it is specified that only 4 .cpp files will be there, and the names of the files have also been given. What if the number of .cpp files varied from test case to test case, then can you do it?