7. This problem asks you to design a Makefile and version-control scheme for automatically generating documentation for Java code.

   Scenario:

   - Assume `a.java` defines one class `A`, and `b.java` defines one class `B`.
   - The `javadoc` program takes a Java file (e.g., `a.java`) that defines a class and makes an HTML file that describes the class (e.g., `a.html`).
   - You need to add a license agreement to the top of every HTML file that `javadoc` produces. The contents of the license are in a file `license`. You have written a shell-script `add-license` that takes an HTML file and changes it so it includes the contents of `license`.

   (a) **8pts** Write a `Makefile` with targets for making `a.html` and `b.html`. The generated files should include the license. They should be remade whenever and only whenever a file that could affect their contents has changed.

   (b) **4pts** Which of the files mentioned in this problem would you put in a version-control system? Briefly justify your inclusion or exclusion of each file.

3. (**15** points)   Write a `Makefile` for this scenario:

- An application `myprog` is written in C, with all the code in `myprog.c`.
- You wrote two test-inputs, in files `input1` and `input2`.
- You want to run `myprog` *with profiling* on each test-input and then use `gprof`, saving the result to file `prof1` (for input `input1`) or `prof2` (for input `prof2`).
- You have a bash script `compare` that takes as arguments two files created by `gprof` and produces an interesting summary. You want a phony `run` target that runs `compare` on `prof1` and `prof2`.

Your `Makefile` should re-compile or re-run programs only as necessary (except the `run` target should always execute `compare`), but it should never use out-dated programs.

Hints: You should have 4 targets. Some will need multiple commands. Some will need multiple sources.