To make it easier to build large programs, UNIX originated the concept of the ***makefile***, a file containing the information necessary to build a program. A makefile not only lists the files that are part of the program, but also describes ***dependencies*** among the files. Suppose that the file `foo.c` includes the file `bar.h`. We say that `foo.c` "depends" on `bar.h`, because a change to `bar.h` will require us to recompile `foo.c`.

Here's a UNIX makefile for the `justify` program. The makefile uses GCC for compilation and linking:

```
justify: justify.o word.o line.o
        gcc -o justify justify.o word.o line.o

justify.o: justify.c word.h line.h
        gcc -c justify.c

word.o: word.c word.h
        gcc -c word.c

line.o: line.c line.h
        gcc -c line.c
```

There are four groups of lines; each group is known as a ***rule***. The first line in each rule gives a ***target*** file, followed by the files on which it depends. The second line is a ***command*** to be executed if the target should need to be rebuilt because of a change to one of its dependent files. Let's look at the first two rules; the last two are similar.

In the first rule, `justify` (the executable file) is the target:

```
justify: justify.o word.o line.o
        gcc -o justify justify.o word.o line.o
```

The first line states that `justify` depends on the files `justify.o`, `word.o`, and `line.o`; if any one of these three files has changed since the program was last built, then `justify` needs to be rebuilt. The command on the following line shows how the rebuilding is to be done (by using the `gcc` command to link the three object files).

In the second rule, `justify.o` is the target:

```
justify.o: justify.c word.h line.h
        gcc -c justify.c
```

The first line indicates that `justify.o` needs to be rebuilt if there's been a change to `justify.c`, `word.h`, or `line.h`. (The reason for mentioning `word.h` and `line.h` is that `justify.c` includes both these files, so it's potentially affected by a change to either one.) The next line shows how to update `justify.o` (by recompiling `justify.c`). The `-c` option tells the compiler to compile `justify.c` into an object file but not attempt to link it.

Q&A Once we've created a makefile for a program, we can use the `make` utility to build (or rebuild) the program. By checking the time and date associated with each