Extra help in typing the GOODBYE.C source code

The first line looks like this:

```
#include <stdio.h>
```

Type a pound sign (press Shift+#) and then **include** and a space. Type a left angle bracket (it's above the comma key) and then **stdio**, a period, **h**, and a right angle bracket. Everything must be in lowercase — no capitals! Press Enter to end this line and start the second line.

Press the Enter key alone on the second line to make it blank. Blank lines are common in programming code; they add space that separates pieces of the code and makes it more readable. And, trust me, anything that makes programming code more readable is okay by me!

Type the word **int**, a space, **main**, and then two parentheses hugging nothing:

```
int main()
```

There is no space between `main` and the parentheses and no space inside the parentheses. Press Enter to start the fourth line.

Type a left curly brace:

```
{
```

This character is on a line by itself, right at the start of the line. Press Enter to start the fifth line.

```
printf("Goodbye, cruel
   world!\n");
```

If your editor was smart enough to automatically indent this line, great. If not, press the Tab key to indent. Then type **printf**, the word *print* with a little *f* at the end. (It's pronounced "print-eff.") Type a left parenthesis. Type a double quote. Type **Goodbye, cruel world**, followed by an exclamation point. Then type a backslash, a little **n**, double quotes, a right parenthesis, and, finally, a semicolon. Press Enter to start the sixth line.

```
return(0);
```

If the editor doesn't automatically indent the sixth line, press the Tab key to start the line with an indent. Then type **return**, a paren, **0** (zero), a paren, and a semicolon. Press Enter.

On the seventh line, type the right curly brace:

```
}
```

Some editors automatically unindent this brace for you. If not, use your editor to back up the brace so that it's in the first column. Press the Enter key to end this line.

Leave the eighth line blank.

The compiler and the linker

After the source code is created and saved to disk, it must be translated into a language the computer can understand. This job is tackled by the compiler.

The *compiler* is a special program that reads the instructions stored in the source code file, examines each instruction, and then translates the information into the machine code understood only by the computer's microprocessor.