Another silly command-prompt program

To see how `puts()` works, create the following program, STOP.C. Yeah, this program is really silly, but you're just starting out, so bear with me:

```
#include <stdio.h>

int main()
{
    puts("Unable to stop: Bad mood error.");
    return(0);
}
```

Save this source code to disk as STOP.C. Compile it, link it, run it.

This program produces the following output when you type **stop** or **./stop** at the command prompt:

```
Unable to stop: Bad mood error.
```

Ha, ha.

- `puts()` is not pronounced "putz."
- Like `printf()`, `puts()` slaps a string of text up on the screen. The text is hugged by double quotes and is nestled between two parentheses.
- Like `printf()`, `puts()` understands escape sequences. For example, you can use `\"` if you want to display a string with a double quote in it.
- You don't have to put a `\n` at the end of a `puts()` text string. `puts()` always displays the newline character at the end of its output.
- If you want `puts()` not to display the newline character, you must use `printf()` instead.

`puts()` and `gets()` in action

The following program is a subtle modification to INSULT1.C. This time, the first `printf()` is replaced with a `puts()` statement:

```
#include <stdio.h>

int main()
{
    char jerk[20];
```