Will the dual variable BOMBER.C program bomb?

Modify the source code for BOMBER.C in your editor. Change the `main()` function so that it reads:

```
int main()
{
    char x;

    printf("Press Enter to drop the bomb:");
    x=getchar();
    dropBomb();
    printf("Key code %d used to drop bomb.\n",x);
    return(0);
}
```

What you're doing is creating another *x* variable for use in the `main()` function. This variable operates independently of the *x* variable in the `dropBomb()` function. To prove it, save your changes to disk and then compile and run the program.

The output is roughly the same for this modified version of the program; the key code displayed for the Enter key "character" is either `10` or `13`, depending on your computer.

- See how the two *x* variables don't confuse the computer? Each one works off by itself.
- Variables in different functions can share the same name.
- For example, you could have a dozen different functions in your program and have each one use the same variable names. No biggie. They're all independent of each other because they're nestled tightly in their own functions.
- Variable *x* not only is used in two different functions — independently of each other — but also represents two different types of variable: a character and an integer. Weird.

Adding some important tension

Face it: You're going nowhere fast as a game programmer with BOMBER.C. What you need is some *tension* to heighten the excitement as the bomb