```
   Print d,c
End
```

### Integer Variable Types

In Igor Pro 7 or later you can use the integer types `int`, `int64` and `uint64` for parameters and local variables in user-defined functions. You can also use `int64` and `uint64` in structures. See **Integer Parameters** on page IV-33 for details.

### Integer Expressions in User-Defined Functions

Prior to Igor Pro 7, all calculations were performed in double-precision floating point. In Igor Pro 7 or later, Igor uses integer calculations when the destination is an integer type. See **Expression Evaluation** on page IV-38 for details.

### Inline Parameters in User-Defined Functions

In Igor Pro 7 or later you can declare user-defined functions parameters inline. See **Inline Parameters** on page IV-33 for details.

### Line Continuation in User-Defined Functions

In user-defined functions in Igor Pro 7 or later, you can use arbitrarily long expressions by including a line continuation character at the very end of a line. See **Line Continuation** on page IV-35 for details.

### Bit Shift Operators in User-Defined Functions

Igor Pro 7 or later supports the bit shift operators << and >> on local variables. See **Bit Shift Operators** on page IV-43 for details.

### Increment and Decrement Operators in User-Defined Functions

In a user-defined function in Igor Pro 7 or later, you can use increment and decrement operators on local variables. See **Increment and Decrement Operators** on page IV-43 for details.

## Legacy Code Issues

This section discusses changes that have occurred in Igor programming over the years. If you are writing new code, you don't need to be concerned with these issues. If you are working with existing code, you may run into some of them.

If you are just starting to learn Igor programming, you have enough to think about already, so it is a good idea to skip this section. Once you are comfortable with the modern techniques described above, come back and learn about these antiquated techniques.

### Old-Style Comments and Compatibility Mode

In Igor Pro 1.00 through 3.00, the comment symbol was the vertical bar (|). Starting with Igor Pro 4.00 vertical bar is used as the bitwise OR operator (see Operators) and the comment symbol is //.

In the unlikely event that you encounter an ancient procedure file that used vertical bar as the comment symbol, you need to replace the vertical bars with //.

From Igor Pro 4.00 through Igor Pro 6.37, there was a compatibility mode that told Igor to accept vertical bar as a comment symbol:

```
Silent 100        // Enter compatibility mode
```

Support for this compatibility mode was removed in Igor Pro 7.00.