

Format Specifiers Print Only the Expression Values

If you want spacing, include it in the format.

Example:

```
printf("%d%d%d", 12, -34, 56);
```

prints

```
12-3456
```

Except for format specifiers and special ASCII characters like linefeed, **characters print exactly as they appear.**

Pitfall: Passing the Wrong Type of Expression

Be sure that your expressions (and ordering) match the format.

Example:

```
printf("%d %f", 10.0, 17);
```

may print (output is system dependent)

```
0 0.000000
```

A C compiler **may be able to warn you** about this kind of error.

Pitfall: Too Few/Many Expressions

If you pass more expressions than format specifiers, **the last expressions are ignored.**

If you pass fewer expressions than format specifiers, **printf prints ... bits!** (In other words, behavior is unspecified.)

Again, a C compiler **may be able to warn you** about this kind of error.

Read Input Using `scanf`

To read values from the keyboard, use `scanf`.

The “f” again means “formatted.”

`scanf` also takes

- a format in quotation marks, and
- a comma-separated list of variable addresses

Example: `int A; scanf ("%d", &A);`

reads a decimal integer, converts it to **2's complement**, and stores the bits in **A**.