- No, `sqrt()` isn't what the Roman legions paraded on their standards. (That was SPQR, which stands for *Senatus Populus Que Romanus,* the Senate, and People of Rome.)
- A reader once wrote me e-mail asking whether the C language had some equivalent of the mathematical *i* dingus, used to represent the *imaginary number* $\sqrt{-1}$, or the square root of "negative one." Because I don't know everything, I had to say that I don't know. Some mathematical C language library somewhere may deal with *i*. But, as far as any other workaround is concerned, I have no idea — though I believe it can be worked into the C++ programming language. (But I don't do C++, so I can't confirm it.)

Strange Math? You Got It!

Most C language libraries are just bursting with math functions. Lots of them. I have listed some of the more common ones in Table 25-1, along with their formats. Pretty much all of them want a `double` or `float` value, which makes sense when you figure that if math had no decimals, more of us would enjoy it.

Table 25-1 Weirdo Math Functions You Never Use

Function	*What It Computes*	*Format*	*Include*	*Library*
`abs`	Absolute value	a=abs(b)	STDLIB.H	standard
`acos`	Arc cosine	x=acos(y)	MATH.H	libm
`asin`	Arc sine	x=asin(y)	MATH.H	libm
`atan`	Arc tangent	x=atan(y)	MATH.H	libm
`cos`	Cosine	x=cos(y)	MATH.H	libm
`exp`	Exponential	x=exp(y)	MATH.H	libm
`log`	Natural logarithm	x=log(y)	MATH.H	libm
`log10`	Base 10 logarithm	x=log10(y)	MATH.H	libm
`sin`	Sine	x=sin(y)	MATH.H	libm
`tan`	Tangent	x=tan(y)	MATH.H	libm

- In Table 25-1, variables *a, b,* and *c* denote integer values. Variables *x, y,* and *z* are doubles.
- The `libm` library is needed only for compiling programs under a Unix-like operating system. Refer to the earlier sidebar "Gotta link in that math library!"