complex numbers. The following program, which uses several <complex.h> functions, computes and displays the roots.

quadratic.c

```
/* Finds the roots of the equation 5x**2 + 2x + 1 = 0 */

#include <complex.h>
#include <stdio.h>

int main(void)
{
  double a = 5, b = 2, c = 1;
  double complex discriminant_sqrt = csqrt(b * b - 4 * a * c);
  double complex root1 = (-b + discriminant_sqrt) / (2 * a);
  double complex root2 = (-b - discriminant_sqrt) / (2 * a);

  printf("root1 = %g + %gi\n", creal(root1), cimag(root1));
  printf("root2 = %g + %gi\n", creal(root2), cimag(root2));

  return 0;
}
```

Here's the output of the program:

```
root1 = -0.2 + 0.4i
root2 = -0.2 + -0.4i
```

The quadratic.c program shows how to display a complex number by extracting the real and imaginary parts and then writing each as a floating-point number. printf lacks conversion specifiers for complex numbers, so there's no easier technique. There's also no shortcut for reading complex numbers; a program will need to obtain the real and imaginary parts separately and then combine them into a single complex number.

27.5 The <tgmath.h> Header (C99): Type-Generic Math

The <tgmath.h> header provides parameterized macros with names that match functions in <math.h> and <complex.h>. These ***type-generic macros*** can detect the types of the arguments passed to them and substitute a call of the appropriate version of a <math.h> or <complex.h> function.

In C99, there are multiple versions of many math functions, as we saw in Sections 23.3, 23.4, and 27.4. For example, the sqrt function comes in a double version (sqrt), a float version (sqrtf), and a long double version (sqrtl), as well as three versions for complex numbers (csqrt, csqrtf, and csqrtl). By using <tgmath.h>, the programmer can simply invoke sqrt without having to worry about which version is needed: the call sqrt(x) could be a call of any of the six versions of sqrt, depending on the type of x.