Computer Exercise 2.3.1

The following program will consist of a rewritten version of *Tri* using four arrays instead of five (the way it is written in the textbook). It will then test the new algorithm on both a nonsymmetric and symmetric tridiagonal system.

```
%seed for random number generator
format default
rng('default')
s = rng;

%nonsymmetric
n = randi([3, 6]);
a = randi([4, 20], n, 1);
d = randi([4, 20], n, 1);
c = randi([4, 20], n, 1);
b = randi([4, 20], n, 1);

b = tri(n, a, d, c, b)
```

```
b = 6×1
   66.8656
  -70.7710
    9.4103
   40.0701
  -36.4742
    9.8685
```

```
%symmetric
a = randi([4, 20], n, 1);
d = randi([4, 20], n, 1);
c = a;
b = randi([4, 20], n, 1);

b = tri(n, a, d, c, b)
```

```
b = 6×1
    0.6773
    0.3932
    0.3761
   -0.3795
    0.6091
    0.8897
```

Here, array "a" and array "c" are equal.

```
function b = tri(n, a, d, c, b)
   for i = 2:n
       xmult = a(i-1)/d(i-1);
       d(i) = d(i) - (xmult*c(i-1));
       b(i) = b(i) - (xmult*b(i-1));
   end
   b(n) = b(n)/d(n);
```

```
    for i = (n-1):-1:1
        b(i) = (b(i) - c(i)*b(i+1))/d(i);
    end
end
```