The resulting code is:

```
C  OBTAINS SOLUTIONS OF THE EQUATION A*X**2 + B*X + C = 0
C
   10 READ(5,11) A, B, C
   11    FORMAT (3F10.0)
      WRITE(6,12) A, B, C
   12    FORMAT ('0A =', 1PE16.6, ', B =', 1PE16.6, ', C =', 1PE16.6)
      IF (A .EQ. 0.0 .AND. B .EQ. 0.0 .AND. C .EQ. 0) STOP
      IF (A .NE. 0.0 .OR. B .NE. 0.0) GOTO 20
         WRITE(6,13) C
   13       FORMAT (' EQUATION SAYS', 1PE16.6,' = 0')
         GOTO 90
   20 IF (A .NE. 0.0) GOTO 30
         R1 = -C/B
         WRITE(6,21) R1
   21       FORMAT (' ONE ROOT. R =', 1PE16.6)
         GOTO 90
C A IS NOT ZERO
   30 IF (C .NE. 0.0) GOTO 40
         R1 = -B/A
         R2 = 0.0
         WRITE(6,31) R1, R2
   31       FORMAT (' R1 =', 1PE16.6, ', R2 =', 1PE16.6)
         GOTO 90
C GENERAL CASE: A, C NON-ZERO
   40 RREAL = -B/(2.0*A)
      DISC = B**2 - 4.0*A*C
      RIMAG = SQRT(ABS(DISC))/(2.0*A)
      IF (DISC .LT. 0.0) GOTO 50
         R1 = RREAL + RIMAG
         R2 = RREAL - RIMAG
         WRITE(6,31) R1, R2
         GOTO 90
   50 R1 = -RIMAG
      WRITE(6,51) RREAL, RIMAG, RREAL, R1
   51    FORMAT (' R1 = (', 1PE16.6, ',', 1PE16.6, ')',
     $          ', R2 = (', 1PE16.6, ',', 1PE16.6, ')')
   90 GOTO 10
      END
```

Two roots are printed only when there are two, one is printed when and only when there is one, and an imaginary part is printed only when called for. More important, it is easy to determine how the program gets to each separate case. (This program is still far from being a general-purpose quadratic-equation solver; the defenses needed against every conceivable numerical hazard require more analysis than we can go into here.)

Write first in an easy-to-understand pseudo-language; then translate into whatever language you have to use.

IF-ELSE constructions formed the framework of our quadratic routine, as they do in most programs. But mere use of IF-ELSE does not guarantee that the result