Listing 1: LU factorization Doolittle

```
program main
    implicit none
    integer, parameter :: n=3
    real*4 a(n,n), c(n,n)
    integer i,j
    ! Llenar la  matrix a
    data (a(1,i), i=1,3) /  80.0, -20.0,  -20.0 /
    data (a(2,i), i=1,3) / -20.0,  40.0,  -20.0 /
    data (a(3,i), i=1,3) / -20.0, -20.0, 130.0 /


    call inversa(a,c,n)

    print*,('-',i=1,79)
    print '(30X,"Inversa de una Matriz")'
    print*,('-',i=1,79)
    print*,''

    !-----------------------------------------
    ! Mostrar la matriz a
    print*, "Matriz a"
    print '(/3f12.6/)',((a(i,j),i=1,n),j=1,n)
    !-----------------------------------------


    print*,''
    print*,''

    !-----------------------------------------
    ! Mostrar la matriz inversa de a
    print*, "Inversa de a"
    print '(/3f12.6/)',((c(i,j),i=1,n),j=1,n)
    !-----------------------------------------
end program main

subroutine inversa(a,c,n)
implicit none
integer n
real*4 a(n,n), c(n,n)
real*4 L(n,n), U(n,n), b(n), d(n), x(n)
real*4 coeff
integer i, j, k

L=0.0
U=0.0
b=0.0

do k=1, n-1
    do i=k+1,n
        coeff=a(i,k)/a(k,k)
        L(i,k) = coeff
        do j=k+1,n
            a(i,j) = a(i,j)-coeff*a(k,j)
        end do
    end do
end do
```

```
57
58 do i=1,n
59     L(i,i) = 1.0
60 end do
61
62 do j=1,n
63     do i=1,j
64         U(i,j) = a(i,j)
65     end do
66 end do
67
68 do k=1,n
69     b(k)=1.0
70     d(1) = b(1)
71
72     do i=2,n
73         d(i)=b(i)
74         do j=1,i-1
75             d(i) = d(i) - L(i,j)*d(j)
76         end do
77     end do
78
79     x(n)=d(n)/U(n,n)
80     do i = n-1,1,-1
81         x(i) = d(i)
82         do j=n,i+1,-1
83             x(i)=x(i)-U(i,j)*x(j)
84         end do
85         x(i) = x(i)/u(i,i)
86     end do
87
88     do i=1,n
89         c(i,k) = x(i)
90     end do
91     b(k)=0.0
92 end do
93 end subroutine inversa
```

Listing 2: Thomas algorithm

```
1  program main
2  !=============================================================
3  !  Solve tridiagonal matrix using thomas algorithm
4  !=============================================================
5  implicit none
6  integer, parameter :: n=7
7  real*4 a(n,3), b(n), x(n)
8  integer i,j,k
9  !-------------------------------------
10 ! Matrix a
11 !-------------------------------------
12 data (a(1,j),j=1,3) / 0.0, -2.25, 1.0 /
13 data (a(2,j),j=1,3) / 1.0, -2.25, 1.0 /
14 data (a(3,j),j=1,3) / 1.0, -2.25, 1.0 /
15 data (a(4,j),j=1,3) / 1.0, -2.25, 1.0 /
16 data (a(5, j),j=1,3) / 1.0, -2.25, 1.0 /
17 data (a(6, j),j=1,3) / 1.0, -2.25, 1.0 /
```

```
18 data (a(7,j),j=1,3)  / 1.0, -2.25, 0.0 /
19 !-----------------------------------------
20 ! Vector b
21 !-----------------------------------------
22 data (b(i),i=1,7) / 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -100.0 /
23     print*,"Matriz a"
24     do i=1,n
25         print*,(a(i,j),j=1,3)
26     end do
27
28     print*,"Vector b"
29     do i=1,n
30         print*,b(i)
31     end do
32
33     call thomas (n,a,b,x)
34     print*,''
35     print*,''
36     print*,"Resultados"
37     print*,"Matriz a'"
38     do i=1,n
39         print*,(a(i,j),j=1,3)
40     end do
41
42     print*,''
43
44     print*,"Vector b'"
45     do i=1,n
46         print*,b(i)
47     end do
48
49     print*,''
50
51     print*,"Vector x"
52     do i=1,n
53         print*,x(i)
54     end do
55 end program main
56 !-----------------------------------------------
57 ! Soubroutine Thomas algorithm
58 !-----------------------------------------------
59 subroutine thomas (n,a,b,x)
60      real*4 a(n,3), b(n), x(n)
61 !----forward elimination----------
62      do i=2,n
63           em=a(i,1)/a(i-1,2)
64           a(i,1)=em
65           a(i,2)=a(i,2) -em*a(i-1,3)
66           b(i)=b(i) -a(i,1)*b(i-1)
67      end do
68 !----back subscitution
69      x(n)=b(n)/a(n,2)
70      do i=n-1,1,-1
71           x(i)= (b(i) -a(i,3)*x(i+1))/a(i,2)
72      end do
73      return
74 end subroutine thomas
```