

```

negat      movem.l   A0/D0-D3,-(SP)  salvataggio registri
          subq.w    #1,D0
          subq.w    #1,D1
          move.w    D0,D3           D3=contatore ciclo righe
loop      move.w    D1,D2           D2=contatore ciclo elementi riga
test      tst.b     (A0)+          verifica elemento riga considerata
          dblt     D2, test        altrimenti prossimo elemento
          bge.s    exit           se valore diverso da 0 terminiamo
          adda.w   D2,A0          A0 va a puntare al primo elemento
                                   della riga successiva a quella da
                                   considerare
          dbra     D3, loop        esaminare riga seguente
exit      movem.l   (SP)+,A0/D0-D3  ripristino registri
          rts

```

```

triang    movem.l   A0/A1/D0/D2,-(SP)  salvataggio registri
          move.w    D0,D2
          subq.w    #2,D0           D0 contatore iterazione esterna
nrow      move.w    D0,D1           D1 contatore iterazione interna
          move.l    A0,A1           Si verifica se
          adda.l    D2,A1           A[i,n]=A[i,n-1]=...=A[i,i+1]=0
test      tst.b     -(A1)         Inizio iterazione interna
          dbne     D1,test        finché j>i e A[i,j]=0
          bne      notrng        se j>i A non è triangolare
          adda.l    D2,A0          altrimenti riga successiva
          dbra     D0, nrow
          move.l    #1, D1         A è triangolare
          bra      fine
notrng    clr.l     D1
fine      movem.l   (SP)+,A0/A1/D0/D2  ripristino registri
          rts

```