

## Laboratorio di Architetture degli elaboratori n. 14, 18-19/4/02

### ESERCIZIO 1

Si progetti una subroutine assembler 68000 che accetti in ingresso:

- 1) un numero positivo  $n$  senza segno a 16 bit nella word meno significativa del registro D0,
- 2) l'indirizzo base di una matrice di  $n \times n$  numeri interi a 8 bit, organizzata per righe, mediante il registro A0.

La subroutine trasforma la matrice nella sua trasposta.

### ESERCIZIO 2

Si progetti una subroutine assembler 68000 che accetti in ingresso:

- 1) un numero positivo  $n$  senza segno a 16 bit nella word meno significativa del registro D0,
- 2) l'indirizzo base di una matrice di  $n \times n$  numeri interi a 8 bit, organizzata per righe, mediante il registro A0.

La subroutine assegna al byte meno significativo di D0 il valore -1 se la matrice e' simmetrica, un valore non negativo altrimenti.

```
trasp  movem.l  A0-A3/D0-D3,-(SP)  salvataggio registri
      move.w   D0,D2                D0=n -> D2
      move.w   D0,D3                D0=n -> D3
      mulu.w   D3,D3                n*n -> D3
      subq.w   #2,D0                D0 contatore iterazione esterna
      move.l   A0,A2                A2:=A0
      adda.l   D3,A2                A2:=A0+(n*n)
      suba.l   D2,A2                A2:=A0+(n-1)*n { (A2)=A[n,1] }
newrow move.w   D0,D1                D1 contatore iterazione interna
      move.l   A0,A1
      adda.l   D2,A1                (A0)=A[i,1]? allora (A1)=A[i+1,1]
      movea.l  A2,A3                se (A2)=A[n,i] allora (A3)=A[n,i]
scamb  move.b  -(A1),D3              | scambio A[i,j] con A[j,i]
      move.b  (A3),(A1)              |
      move.b  D3,(A3)                |
      suba.l  D2,A3                (A3)=A[n,i]? allora (A3)=A[n,i-1]
      dbra   D1,scamb              esegui nuovo test
      adda.l  D2,A0                (A0)=A[i,1]? allora (A0)=A[i+1,1]
      adda.l  #1,A2                (A2)=A[n,i]? allora (A2)=A[n,i+1]
      dbra   D0,newrow
fine  movem.l  (SP)+,A0-A3/D0-D3  ripristino registri
      rts
```

```

simm  movem.l A0-A3/D0-D3,-(SP) salvataggio registri
      move.w D0,D2          D0=n -> D2
      move.w D0,D3          D0=n -> D3
      mulu.w D3,D3          n*n -> D3
      subq.w #2,D0          D0 contatore iterazione esterna
      move.l A0,A2          A2:=A0
      adda.l D3,A2          A2:=A0+(n*n)
      suba.l D2,A2          A2:=A0+(n-1)*n { (A2)=A[n,1] }
newrow move.w D0,D1          D1 contatore iterazione interna
      move.l A0,A1
      adda.l D2,A1          (A0)=A[i,1]? allora (A1)=A[i+1,1]
      movea.l A2,A3          se (A2)=A[n,i] allora (A3)=A[n,i]
test  move.b -(A1),D3       | confronta A[i,j] con A[j,i]
      cmp.b (A3),D3         |
      suba.l D2,A3          (A3)=A[n,i]? allora (A3)=A[n,i-1]
      dbne D1,test         esegui nuovo test
      bne nonsim           NO: A non è simmetrica
      adda.l D2,A0          (A0)=A[i,1]? allora (A0)=A[i+1,1]
      adda.l #1,A2          (A2)=A[n,i]? allora (A2)=A[n,i+1]
      dbra D0,newrow
      move.l #1,D4
      bra fine
nonsim clr.l D4
fine  movem.l (SP)+,A0-A3/D0-D3 ripristino registri
      rts

```