Esercizio di programmazione

sino a 12 punti – è possibile consultare solamente l'instruction set Intel - tempo: 60 minuti

Sono date due matrici quadrate contenenti numeri con segno su 16 bit, memorizzate per righe, di `DIM`x`DIM` elementi. Si scriva una procedura `variazione` in linguaggio Assembly 8086 in grado di calcolare la variazione percentuale (troncata all'intero) tra gli elementi di indice corrispondente della *riga I* della prima matrice ([*I*, 0], [*I*, 1], [*I*, 2]…) e della *colonna I* della seconda ([0, *I*], [1, *I*], [2, *I*]…). Ad esempio, nel caso di due matrici 3x3 e con *I* = 2:

$$\begin{bmatrix} 4 & -45 & 15565 \\ 6458 & 4531 & 124 \\ -548 & 2124 & 31000 \end{bmatrix} \qquad \begin{bmatrix} 6 & -5421 & -547 \\ -99 & 4531 & 1456 \\ 4592 & 118 & 31999 \end{bmatrix}$$

il risultato è 0, -31, 3.

La variazione percentuale è calcolata come segue:

$$Variazione = (Val2 - Val1) \cdot 100 \,/\, Val1$$

La procedura riceve l'indirizzo delle due matrici e l'indice *I* mediante *stack*, mentre fornisce i risultati sulla porta C di un modulo Intel 8255 collegato al processore e accessibile a partire dall'indirizzo 0x80h, da considerarsi già programmato in modo 0-output per i gruppi A e B.
Non si devono usare variabili aggiuntive, né modificare i dati presenti in memoria.
Di seguito un esempio di programma chiamante.

```
DIM EQU 3

[…]

.data
mat1 dw    4,   -45, 15565
     dw 6458,  4531,   124
     dw -548,  2124, 31000
mat2 dw    6, -5421,  -547
     dw  -99,  4531,  1456
     dw 4592 ,  118, 31999

.code

[…]

LEA AX, MAT1
LEA BX, MAT2
PUSH AX
PUSH BX
PUSH 2

call variazione
add sp, 6

[…]
```

Soluzione proposta

```
porta   EQU 80h
portb   EQU porta+1
portc   EQU portb+1
control EQU portc+1

DIM     EQU 3

#start=8255.exe#

.model  small
.stack
.data
mat1    DW    4,   -45, 15565
        DW 6458,  4531,   124
        DW -548,  2124, 31000
mat2    DW    6, -5421,  -547
        DW  -99,  4531,  1456
        DW 4592 ,  118, 31999

.code
.startup

        MOV DX, control
        MOV AL, 10000000b
        OUT DX, AL

        LEA AX, MAT1
        LEA BX, MAT2
        PUSH AX
        PUSH BX
        PUSH 2

        CALL variazione
        ADD SP, 6

.exit


proc    variazione
        PUSH BP
        MOV BP, SP
        PUSH AX
        PUSH BX
        PUSH CX
        PUSH DX
        PUSH SI
        PUSH DI

        MOV AX, [BP+4]
        MOV SI, [BP+8] ; MAT1
        MOV BX, 2*DIM
        MUL BX
        ADD SI, AX
        MOV AX, [BP+4]
        MOV DI, [BP+6] ; MAT2
        SHL AX, 1
        ADD DI, AX
```

```
        MOV CX, DIM

ciclo:  MOV AX, [DI] ; assumo che non si abbia overflow nella sottrazione
        SUB AX, [SI] ; (altrimenti potrei fare un salto condizionato su overflow flag)

        MOV BX, 100
        IMUL BX
        MOV BX, [SI]
        IDIV BX
        OUT portc, al ; assume che la variazione perc. sia compresa tra -128 e +127

        ADD SI, 2
        ADD DI, 2*DIM
        LOOP ciclo

        POP DI
        POP SI
        POP DX
        POP CX
        POP BX
        POP AX
        POP BP
        RET
endp    variazione

        end
```