

Esercizio di programmazione

sino a 12 punti – è possibile consultare qualunque materiale cartaceo - tempo: 60 minuti

Data una matrice di interi positivi memorizzata per righe, si scriva in linguaggio Assembly 8086 una procedura **calcola_e_invia_media** che calcoli la media degli elementi di ciascuna colonna della matrice.

La procedura riceve tramite stack:

- l'offset di una matrice di N righe e M colonne, dove N e M sono dichiarati come costanti
- l'offset di un vettore di M elementi, non inizializzato.

Gli elementi della matrice sono memorizzati come word. Anche il valore medio di ciascuna colonna è memorizzato come word, ma i calcoli devono essere eseguiti su 32 bit per evitare overflow e minimizzare l'errore di approssimazione.

Per ogni colonna della matrice, la procedura deve innanzitutto calcolare la media degli elementi e salvarla nell'elemento corrispondente del vettore (ad esempio, la media della prima colonna della matrice deve essere memorizzata nel primo elemento del vettore).

Successivamente, la procedura deve inviare il contenuto del vettore ad una periferica, che si interfaccia all'8086 tramite il modulo 8255. La procedura deve quindi scrivere tutti gli elementi del vettore (un byte alla volta) sulla porta A dell'8255, che si assume precedentemente configurata in modo 0 output. L'8086 e la periferica utilizzano la stessa modalità di memorizzazione delle word (little endian).

Di seguito un esempio di programma chiamante:

```
N EQU 5      ; numero di righe
M EQU 4      ; numero di colonne
PORTA EQU 80h
PORTB EQU PORTA+1
PORTC EQU PORTA+2
CONTROL EQU PORTA+3

#START=8255.exe#

.MODEL small
.STACK
.DATA
matrice DW 20, 35, 40, 12
         DW 26, 5, 18, 30
         DW 10, 45, 33, 58
         DW 47, 3, 35, 34
         DW 60, 45, 32, 43
media DW M DUP (?)

.CODE
.STARTUP
PUSH OFFSET matrice
PUSH OFFSET media
CALL calcola_e_invia_media
ADD SP, 4
.EXIT
```

Soluzione proposta

; in grassetto la parte richiesta dalla traccia

```
N      EQU 5      ; numero di righe
M      EQU 4      ; numero di colonne
PORTA  EQU 80h
PORTB  EQU PORTA+1
PORTC  EQU PORTA+2
CONTROL EQU PORTA+3
```

#START=8255.exe#

```
        .MODEL small
        .STACK
        .DATA
matrice DW 20, 35, 40, 12
        DW 26, 5, 18, 30
        DW 10, 45, 33, 58
        DW 47, 3, 35, 34
        DW 60, 45, 32, 43
media   DW M DUP (?)

        .CODE
        .STARTUP
MOV DX, CONTROL      ; configurazione 8255
MOV AL, 10000000b
OUT DX, AL
PUSH OFFSET matrice
PUSH OFFSET media
CALL calcola_e_invia_media
ADD SP, 4
.EXIT
```

calcola_e_invia_media PROC

```
        PUSH BP
        MOV BP, SP
        PUSH AX
        PUSH BX
        PUSH CX
        PUSH DI
        MOV CX, M
        MOV BX, [BP + 6]
        MOV DI, [BP + 4]

inizio: XOR AX, AX      ; inizializzazione accumulatore su 32 bit (DX AX)
        XOR DX, DX
        XOR SI, SI     ; indice per riga

somma: ADD AX, [BX][SI] ; somma degli elementi della stessa colonna
        ADC DX, 0
        ADD SI, M * 2
        CMP SI, N * M * 2
        JB somma

        MOV SI, N      ; calcolo della media
        DIV SI
        MOV [DI], AX   ; copia del risultato nel vettore
        ADD BX, 2
        ADD DI, 2
        LOOP inizio
```

```
        MOV CX, 2 * M          ; output su 8255
        MOV DI, [BP + 4]
        MOV DX, PORTA
scrivi: MOV AL, [DI]
        OUT DX, AL
        INC DI
        LOOP scrivi

        POP DI
        POP SI
        POP DX
        POP CX
        POP BX
        POP AX
        POP BP
        RET
calcola_e_invia_media ENDP
END
```