

Esercizio di programmazione

Sino a 12 punti. È possibile consultare solamente l'instruction set Intel fornito. Tempo: 60 minuti

Sia data una matrice byte contenente valori positivi, di dimensione fissata dalle due costanti (strettamente positive) **NRIGHE** e **NCOLONNE**. Si scriva una procedura **contaColonne** in linguaggio Assembly 8086 in grado di contare in quante colonne della matrice la somma degli elementi pari è superiore alla somma degli elementi dispari.

La procedura riceve l'offset della matrice tramite stack e restituisce nel registro AX il numero di colonne in cui la somma degli elementi pari è superiore alla somma degli elementi dispari.

Non è ammesso l'uso di variabili.

Di seguito un esempio di programma chiamante:

```
NRIGHE EQU 4
NCOLONNE EQU 6
.MODEL small
.STACK
.DATA
matrice DB 113, 10, 95, 20, 60, 10
        DB 51, 26, 120, 30, 56, 13
        DB 102, 171, 21, 111, 17, 22
        DB 204, 100, 230, 16, 76, 34

.CODE
.STARTUP
LEA SI, matrice
PUSH SI
CALL contaColonne
.EXIT
```

Nell'esempio, il valore del registro AX dopo la chiamata della procedura **contaColonne** è 4.

Soluzione proposta:

```
NRIGHE      EQU 4
NCOLONNE    EQU 6
.MODEL small
.STACK
.DATA
matrice      DB 113, 10, 95, 20, 60, 10
              DB 51, 26, 120, 30, 56, 13
              DB 102, 171, 21, 111, 17, 22
              DB 204, 100, 230, 16, 76, 34

.CODE

.STARTUP
LEA SI, matrice
PUSH SI
CALL contaColonne
POP AX
.EXIT

PROC        contaColonne
PUSH BP
MOV BP, SP
PUSH BX
PUSH CX
PUSH DX
PUSH SI
MOV SI, [BP+4]
XOR AX, AX      ; contatore risultato
MOV CX, NCOLONNE
cicloC:     PUSH CX
            PUSH AX
            PUSH SI
            MOV CX, NRIGHE
            XOR DX, DX      ; somma elementi dispari
            XOR BX, BX      ; somma elementi pari
cicloR:     MOV AL, [SI]
            XOR AH, AH      ; estensione a 16 bit
            TEST AL, 1
            JZ pari
            ADD DX, AX      ; si suppone che la somma massima sia < 65536
            JMP next
pari:       ADD BX, AX
next:       ADD SI, NCOLONNE
            LOOP cicloR
            POP SI
            POP AX
            POP CX
            CMP BX, DX
            JBE salta
            INC AX
salta:      INC SI
            LOOP cicloC
            POP SI
            POP DX
            POP CX
            POP BX
            POP BP
            RET
ENDP        contaColonne
END
```

Soluzione alternativa:

```
contaColonne PROC
MOV BP, SP
MOV BX, [BP + 2]
MOV DI, 0 ; contatore del numero di colonne con pari > dispari

MOV CX, NCOLONNE
cicloEsterno:
PUSH CX
MOV CX, NRIGHE
MOV SI, 0
MOV AX, 0 ; somma numeri pari
MOV DX, 0 ; somma numeri dispari
cicloInterno:
TEST [BX][SI], 1
JNZ dispari
; numero pari
ADD AL, [BX][SI]
ADC AH, 0
JMP prossimaRiga
dispari:
ADD DL, [BX][SI]
ADC DH, 0
prossimaRiga:
ADD SI, NCOLONNE
LOOP cicloInterno
CMP AX, DX
JB prossimaColonna
INC DI
prossimaColonna:
INC BX
POP CX
LOOP cicloEsterno

MOV AX, DI
RET
contaColonne ENDP
```