

Esercizio di programmazione

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

Sia data una matrice quadrata di byte di dimensione DIMxDIM (DIM dichiarato come costante) memorizzata per righe. Gli elementi della matrice contengono il valore 0 o il valore 1.

Si vuole verificare se la matrice contiene almeno un quadrato 3x3 di 9 elementi contigui aventi valore 1.

Si scriva una procedura **cercaQuadrato** che riceve in input

- in AX l'indice progressivo di un elemento della matrice nel vettore che la memorizza
- in BX l'offset nel data segment del primo elemento della matrice.

La procedura restituisce in output (attraverso il registro DX) uno fra i due valori seguenti:

- 1, se l'elemento il cui indice è stato ricevuto in input corrisponde all'elemento centrale di un quadrato 3x3 di elementi con valore 1
- 0, altrimenti.

Si assuma che tutti gli elementi della prima e ultima colonna, nonché della prima e ultima riga della matrice contengano il valore 0.

Ad esempio, nel caso

```
0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 1, 1, 0, 0, 0
0, 0, 0, 0, 1, 1, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 1, 1, 1, 1, 0, 0, 0
0, 0, 1, 1, 1, 0, 0, 0, 0
0, 0, 1, 1, 1, 0, 0, 0, 0
0, 0, 0, 0, 1, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0
```

la procedura **cercaQuadrato** restituisce 1 se chiamata con AX = 48, 0 in tutti gli altri casi.

Di seguito un esempio di programma che chiama la procedura **cercaQuadrato** per ogni elemento della matrice, finché non è rilevato un quadrato:

```
[...]
XOR AX, AX ; indice dell'elemento corrente
LEA BX, matrice
inizioCiclo:
CALL cercaQuadrato
CMP DX, 1
JE fineCiclo
INC AX
CMP AX, DIM * DIM
JL inizioCiclo
fineCiclo:
[...]
```

Soluzione

```
DIM      EQU 9
.model small
.stack
.data

matrice  db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
          db 0, 0, 0, 0, 1, 1, 0, 0, 0, 0
          db 0, 0, 0, 0, 1, 1, 0, 0, 0, 0
          db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
          db 0, 0, 1, 1, 1, 1, 0, 0, 0, 0
          db 0, 0, 1, 1, 1, 0, 0, 0, 0, 0
          db 0, 0, 1, 1, 1, 0, 0, 0, 0, 0
          db 0, 0, 0, 0, 1, 0, 0, 0, 0, 0
          db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

          .code
          .startup
          XOR AX, AX          ; indice dell'elemento corrente
          LEA BX, matrice
inizioCiclo: CALL cercaQuadrato
              CMP DX, 1
              JE fineCiclo
              INC AX
              CMP AX, DIM*DIM
              JL inizioCiclo
fineCiclo:
          .exit
```

cercaQuadrato proc

```
          PUSH BX
          PUSH CX

          XOR DX, DX
          ADD BX, AX
          CMP [BX], 1
          JNE fine

          MOV DX, 1
          SUB BX, DIM+1
          MOV CX, 3
loopext:  PUSH CX
          MOV CX, 3
loopint:  CMP [BX], 1
          JNE no_quadrato
          INC BX
          LOOP loopint

          ADD BX, DIM-3
          POP CX
          LOOP loopext
          JMP fine
```

```

no_quadrato: POP CX
              XOR DX, DX
fine:        POP CX
              POP BX
              RET
cercaQuadrato endp

```

```

cercaQuadrato proc                ; soluzione alternativa

```

```

    PUSH BX
    MOV DX, 1
    ADD BX, AX
    CMP [BX], 1
    JNE no_quadrato
    CMP [BX-DIM-1], 1
    JNE no_quadrato
    CMP [BX-DIM], 1
    JNE no_quadrato
    CMP [BX-DIM+1], 1
    JNE no_quadrato
    CMP [BX-1], 1
    JNE no_quadrato
    CMP [BX+1], 1
    JNE no_quadrato
    CMP [BX+DIM-1], 1
    JNE no_quadrato
    CMP [BX+DIM], 1
    JNE no_quadrato
    CMP [BX+DIM+1], 1
    JNE no_quadrato
    JMP fine

```

```

no_quadrato: XOR DX, DX
fine:        POP BX
              RET
cercaQuadrato endp

```

```

END

```