

Esercizio 3

Si realizzi in assembly MIPS il seguente programma. Sia data in memoria una matrice quadrata di numeri da 2 byte ciascuno di dimensioni $N \times N$, con N non minore di 3.

Il programma assembly deve definire nella sezione dedicata al **data segment** la matrice (da denominarsi **M**) e la dimensione del lato (da denominarsi **N**). Si assume per ipotesi che la matrice **M** sia coerente con la dimensione del lato e che il lato non sia minore di 3 come su menzionato: non occorre dunque effettuare alcun controllo per verificare la sua validità.

Nella sezione dedicata al **text segment**, il programma deve avere nel comparto **main** il caricamento dei dati da memoria, la chiamata ad una funzione **sommaScacchiera** definita di seguito e la stampa a terminale dei risultati di **sommaScacchiera**. La stampa a terminale *non deve avvenire all'interno della funzione* ma nel comparto **main** chiamante.

La funzione **sommaScacchiera** accetta come parametri di **input**:

– \$a0: l'indirizzo base della matrice

– \$a1: il lato della matrice

e restituisce in **output**:

– \$v0: la somma degli elementi nelle righe *pari* che occupano la colonna *pari*

– \$v1: la somma degli elementi nelle righe *dispari* che occupano la colonna *dispari*

(cominciando a contare gli indici di riga e di colonna da 0: la prima colonna e la prima riga sono dunque da considerarsi pari e così via).

Esempi:

a) Input: \$a0 è l'indirizzo in memoria di **M** definita come di seguito, con lato \$a1 che vale $N = 3$

```
0 1 2
3 4 5
6 7 8
```

Output: \$v0 vale $0 + 2 + 6 + 8 = 16$; \$v1 vale 4.

b) Input: \$a0 è l'indirizzo in memoria di **M** definita come di seguito, con lato \$a1 che vale $N = 4$

```
5 6 4 8
2 5 2 6
1 9 8 4
7 3 2 2
```

Output: \$v0 vale $5 + 4 + 1 + 8 = 18$; \$v1 vale $5 + 6 + 3 + 2 = 16$.

Note:

Commentare *ogni riga di codice* avendo cura di spiegare a cosa servano i registri. Una soluzione ricorsiva sarà premiata con un bonus di 1 punto.