

Esercizio 3 (4 punti).

Si realizzi in assembly MIPS il seguente programma. Sia dato in memoria un array di lunghezza N, con N non minore di 2, contenente numeri in formato di 4 byte ciascuno.

Il programma assembly deve definire nella sezione dedicata al **data segment** l'array (da denominarsi M) e la lunghezza (da denominarsi N). Si assume per ipotesi che la lunghezza dell'array non sia minore di 2 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 `sommaContaUgualiPrec` definita di seguito e la stampa a terminale dei risultati di `sommaContaUgualiPrec`. La stampa a terminale *non deve avvenire all'interno della funzione* ma nel comparto main chiamante.

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

– \$a0: l'indirizzo base dell'array;

– \$a1: la lunghezza dell'array;

e restituisce in **output**:

– \$v0: la somma degli elementi di valore uguale all'elemento precedente dell'array;

– \$v1: il conteggio degli elementi di valore uguale all'elemento precedente dell'array.

Esempi:

a) Input: \$a0 è l'indirizzo in memoria di M definito come di seguito, con lunghezza \$a1 che vale N = 5

1, 1, 4, 2, 2

Output: \$v0 vale $1 + 2 = 3$; \$v1 vale 2.

b) Input: \$a0 è l'indirizzo in memoria di M definito come di seguito, con lunghezza \$a1 che vale N = 7

0, 6, 7, 8, 8, 8, 6

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

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.