

1. [10] Si traducano in linguaggio Assembly MIPS nativo (evitando cioè di utilizzare pseudoistruzioni) le seguenti procedure in linguaggio C.

La procedura **SQArray()** modifica tutti gli elementi di un array presente in memoria. La procedura si aspetta l'indirizzo base dell'array **vett** nel registro **\$a0** ed il numero di elementi del vettore **n_elem** in **\$a1**. La procedura **SemiQuad()** si aspetta l'argomento **n** in **\$a0** e restituisce il risultato in **\$v0**.

```
int SQArray(int vett[], int n_elem)      int SemiQuad(int n)
{
    for(int i=0; i<n_elem; i++)          {
        vett[i] = SemiQuad( vett[i] );    return(n*n/2);
    }
}
```

2. [8] Si scriva un programma Assembly completo, per ambiente SPIM, che permetta ad un utente di inserire in memoria dati un array, i cui elementi vengono chiesti all'utente da tastiera. Successivamente, il programma modifica l'array chiamando la procedura **SQArray()** e quindi termina, avvisando che il calcolo è terminato. Il programma deve presentarsi a terminale come nel seguente esempio:

```
Numero di elementi dell'array > 5
Inserisci elemento 0 > 123
Inserisci elemento 1 > 3245
...
Inserisci elemento 4 > 1000
Calcolo terminato.
```

3. [4] Tradurre il seguente frammento di codice: a) in Assembly MIPS nativo e b) in linguaggio macchina (specificando ampiezza in bit e valore dei campi delle istruzioni).

```
lw $t1, $t2($s0) # load word, offset specified in $t2
divi $s2, $s0, 8 # divide by immediate
```

4. [5] Si traduca il seguente frammento di codice Assembly MIPS in linguaggio macchina, in formato esadecimale, calcolando prima i valori esadecimali **N1** e **N2** che permettono di saltare esattamente all'indirizzo indicato in ciascun commento.

```
0x12345678:    beq $s0, $s4, N1      # salta a: 0x12345660
                jal N2      # salta a: 0x10203040
```

5. [6] Rappresentare gli indirizzi ed il contenuto, byte per byte (in formato esadecimale), del segmento dati della memoria che viene modificato a seguito dell'esecuzione delle seguenti direttive (si ricorda che il codice ASCII numerico di "A"=65, "B"=66, ...):

```
.data 0x500
.word -20, 0x500
.asciiz "ABCDE"
.half -5, -10, -15
spazio: .space 12
```

A che indirizzo corrisponde la label **spazio**?:

System calls

	codice (\$v0)	argomenti	risultato
print int	1	\$a0	
print float	2	\$f12	
print double	3	\$f12	
print string	4	\$a0	
read int	5		\$v0
read float	6		\$f0
read double	7		\$f0
read string	8	\$a0, \$a1	
sbrk	9	\$a0	\$v0
exit	10		

Registri MIPS

	0	zero	24-25	t8 - t9
1	at		26-27	k0 - k1
2-3	v0 - v1		28	gp
4-7	a0 - a3		29	sp
8-15	t0 - t7		30	s8
16-23	s0 - s7		31	ra

MIPS Instruction Set:

