

1. [9] Si traducano in linguaggio Assembly MIPS nativo (evitando cioè di utilizzare pseudoistruzioni) le seguenti procedure in linguaggio C.

La procedura **SommaCub()** si aspetta l'indirizzo base dell'array **vett** nel registro **\$a0**, il numero di elementi del vettore **n_elem** in **\$a1** e restituisce il risultato in **\$v0**. La procedura **Cubo()** si aspetta l'argomento **n** in **\$a0** e restituisce il risultato in **\$v0**.

```
int SommaCub(int vett[], int n_elem) {
    int i, sum=0;
    for(i=0; i<n_elem; i++)
        sum = sum + Cubo(vett[i]);
    return( sum );
}

int Cubo(int n)
{
    return( n*n*n );
}
```

2. [9] Si scriva un programma Assembly completo, per ambiente SPIM, che permetta ad un utente di calcolare il valore della funzione "SommaCub" di un array inserito da tastiera dall'utente. Il programma esegue il calcolo chiamando la funzione **SommaCub()** sviluppata nell'esercizio precedente, quindi termina. Il programma deve presentarsi a terminale come nel seguente esempio:

```
Inserire il numero di elementi dell'array > 3
Inserisci elemento 0 > 9
Inserisci elemento 1 > 5
Inserisci elemento 2 > 10
La somma dei cubi vale: 1854
```

3. [4] Tradurre ciascuna delle seguenti pseudoistruzioni: a) in Assembly MIPS nativo e b) in linguaggio macchina (specificando ampiezza in bit e valore dei campi delle istruzioni).

```
bgti $s2, -7, -20 # branch on greater than immediate
divi $t0, $s0, 410 # divide by immediate
```

4. [4] Tradurre in linguaggio macchina MIPS la seguente istruzione e calcolare l'effettivo indirizzo di salto. Determinare inoltre gli indirizzi minimo e massimo a cui tale istruzione, con l'opportuno argomento, sarebbe in grado di saltare.

```
Indirizzo Istruzione
0xC6787200: jal 0x600
```

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 0x300
.word -100, +100, 0x100
.asciiz "ABCDEFGH"
.half -1, -2, -3
```

Se, all'inizio del programma Assembly immediatamente seguente a tali direttive, viene chiamata una system call **sbrk**, quale sarà il valore restituito nel registro **\$v0**?

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:

