1. [10] Si traduca in linguaggio Assembly MIPS nativo (evitando di utilizzare pseudoistruzioni) la seguente procedura in linguaggio C.

La procedura si aspetta l'argomento **n** nel registro **$a0** e restituisce il risultato in **$v0**.

```
int MioFibo( int n )
{
     if( n < 2 )
          return( 1 );
     else
          return( 3*MioFibo(n-1) - MioFibo(n-2) );
}
```

2. [8] Si scriva un programma Assembly completo, per ambiente SPIM, che permetta ad un utente di calcolare il valore della funzione "MioFibo" di un argomento fornito da tastiera da un utente. Il programma esegue il calcolo chiamando la funzione **MioFibo()** sviluppata nell'esercizio precedente, quindi chiede un nuovo argomento. Se l'argomento fornito è 0, il programma termina.

Il programma deve presentarsi a terminale come nel seguente esempio:

```
Inserire il valore dell'argomento
> 4
MioFibo( 4 ) = 13
Inserire il valore dell'argomento
> 0
Programma terminato.
```

3. [7] Si traduca il seguente frammento di codice: a) in Assembly MIPS nativo e b) in linguaggio macchina, specificando valore e ampiezza in bit dei campi di ogni istruzione. Si consideri che il codice viene caricato in memoria a partire dall'indirizzo **0x500**.

```
0x500:  inizio:   lw $t0, $a0($s0)
        ciclo:    li $s5, 4^11
                  bgti $t0, 100, ciclo     # branch if greater than
                  j inizio
```

4. [5] 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:

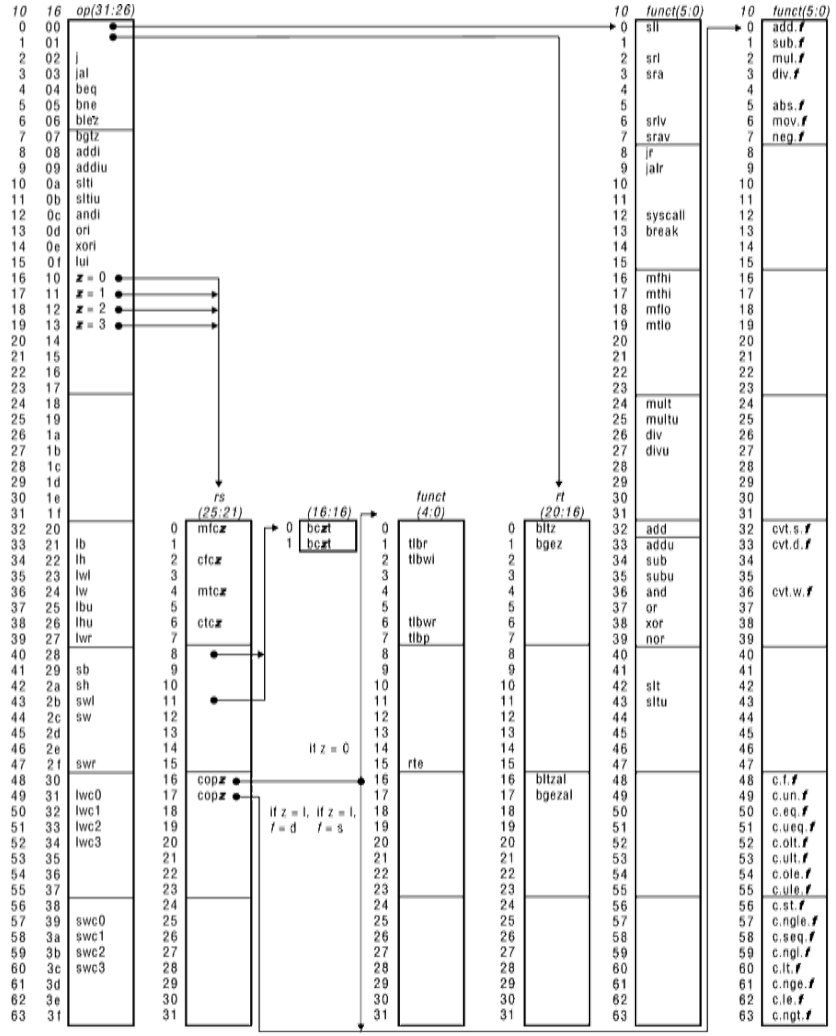
```
.data 1600
.word -500
.space 6
.half 0x300, -6
```

Se, all'inizio del programma Assembly immediatamente seguente a tali direttive, viene chiamata una system call **sbrk**, quale sarà il valore di indirizzo restituito?

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: