1. [2] Che numero decimale rappresenta la parola **0xCC808000** nel formato standard IEEE–754, singola precisione?

2. [3] Si dimostri la seguente equivalenza applicando le regole dell'algebra booleana: $(\bar{a}+\bar{b})(\bar{b}+\bar{c})(a+b+c)=(a+c)\oplus b$

3. [5] Si progetti un circuito caratterizzato da 3 linee di ingresso ($n_2\, n_1\, n_0$) che rappresentano un numero *N* intero con segno (in complemento a 2), e da un'uscita Y che vale '1' se *N* è positivo e dispari, oppure se *N* è negativo e pari.
a) Determinare la tabella di verità di Y; b) esprimerla nella forma canonica più adatta; c) semplificarla mediante mappe di Karnaugh; d) semplificarla ulteriormente, se possibile, mediante semplificazioni algebriche; e) disegnarne lo schema circuitale.

4. [8] Si sintetizzi una macchina a stati finiti di Moore caratterizzata da una linea d'ingresso e una linea di uscita. L'uscita cambia valore ogni qualvolta sulla linea d'ingresso si presenta un fronte di salita seguito immediatamente da un fronte di discesa. Si assuma che, allo stato iniziale, sia l'ingresso che l'uscita siano a '0'.
Si determinino: STG, STT, STT codificata e struttura circuitale del sistema completo, non trascurando la gestione del segnale di clock ed avendo cura di semplificare il più possibile le funzioni prima di tradurle in circuito.

5. [5] Si traduca in Assembly MIPS nativo (evitando di utilizzare pseudo-istruzioni) la seguente procedura in linguaggio C.

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

```
int Product(int vett[], int n_el)
{
    int i;
    int prod=1;
    for(i=0; i<n_el; i++)
        prod = prod * vett[i];
    return( prod );
}
```

6. [5] Si scriva un programma Assembly completo, per ambiente SPIM, che permetta ad un utente di calcolare il valore della funzione "Product" di un array inserito da tastiera dall'utente. Per il calcolo, viene chiamata la funzione **Product()** sviluppata nell'esercizio precedente.

Il programma deve presentarsi a terminale come nell'esempio riportato a lato:

```
Inserire il numero di elementi > 12
Inserisci elemento 0  > 9
Inserisci elemento 1  > 5
...
Inserisci elemento 11  > 1
Product = 12345
```

7. [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 effettivamente il salto all'indirizzo indicato in ciascun commento.

```
0xABCDEF00:   bne $t0, $t1, N1      # salta a: 0xABCDEEDC
              j N2                  # salta a: 0xABCD0000
```


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: