1. [2] Che numero decimale rappresenta, nello standard IEEE–754 singola precisione, il codice: **0x0000000C** (lo si esprima pure mediante potenze di 2)?

2. [2] Esprimere la $f(a,b,c) = a\overline{b} + b\overline{c}$ funzione in seconda forma canonica.

3. [4] Si disegni lo schema di un circuito che calcola la differenza tra due numeri binari con segno di 4 bit ciascuno; si calcoli il cammino critico, evidenziandolo sullo schema circuitale. È possibile che il circuito vada in overflow? Se sì, in quale caso, ad esempio?

4. [5] Si progetti un circuito caratterizzato da quattro ingressi (a_3 a_2 a_1 a_0) sui quali viene presentato un numero binario di 4 bit, intero con segno, e da un'uscita Y che vale '1' se e solo se il numero in ingresso è divisibile per 3 o per 5 (0 è divisibile per entrambi).
a) Determinare le 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) disegnare lo schema del circuito.

5. [7] Si sintetizzi una macchina a stati finiti di Moore sincrona, caratterizzata da una linea d'ingresso I e due linee di uscita y_1 e y_0 che rappresentano un numero binario senza segno. La macchina viene incrementata ad ogni fronte di salita su I e, quando arriva a 2, viene decrementata ad ogni fronte di discesa su I fino a 0, quindi torna ad essere incrementata dai fronti di salita, e così via. Si considerino inizialmente I, y_1 e y_0 tutti a '0'.
Si determinino: STG, STT, STT codificata e struttura circuitale del sistema completo, gestendo il segnale di clock ed avendo cura di semplificare il più possibile le funzioni prima di tradurle in circuito.

6. [6] Si traduca il seguente frammento di codice Assembly MIPS in linguaggio macchina, in formato esadecimale, calcolando prima i valori esadecimali **L1** e **L2** che permettono di saltare esattamente all'indirizzo indicato nel commento.

```
0x3C00:  beq $4, $20, L1        # salta a: 0x3BCC
         jal L2                 # salta a: 0x00224488
```

7. [6] Si scriva un programma Assembly, per ambiente SPIM, che richieda all'utente da tastiera 20 numeri interi, memorizzandoli in un array allocato staticamente, quindi chiami una funzione alla quale passa l'indirizzo base dell'array e dalla quale ottenga in restituzione la somma dei quadrati degli elementi dell'array; infine stampi tale somma e termini. Il programma deve apparire a video come nell'esempio a lato.

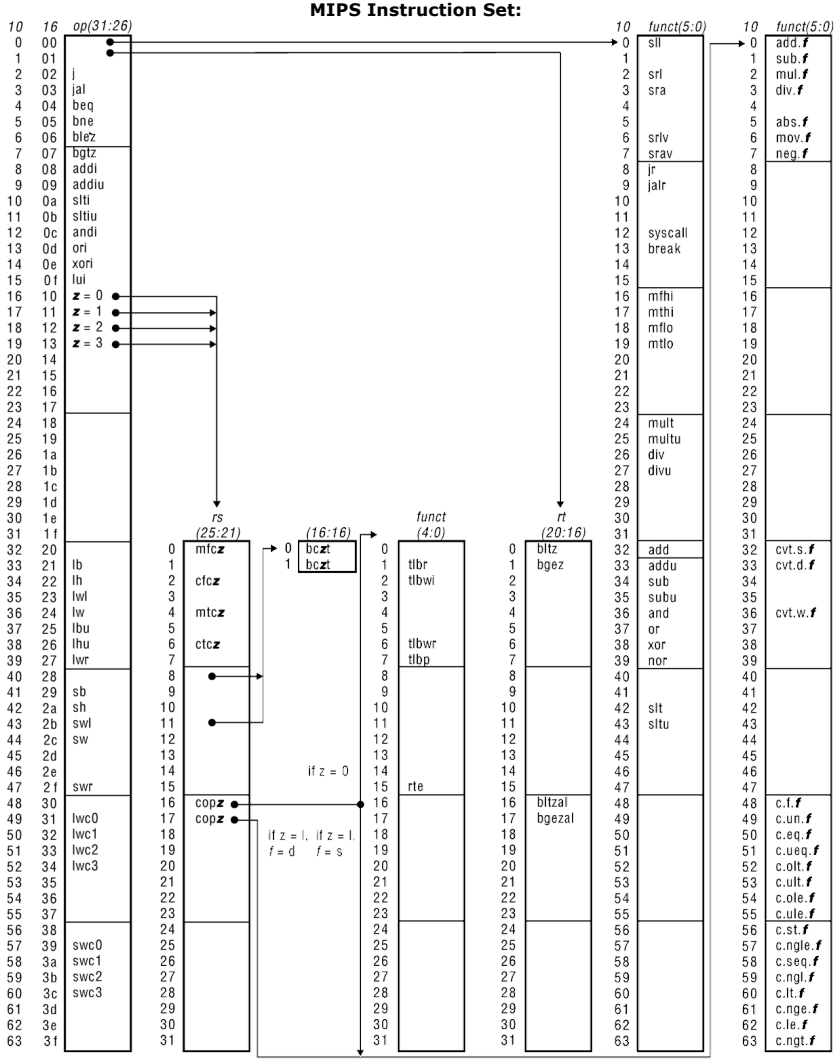
```
Elemento 1 > 5
Elemento 2 > 2
...
Elemento 20 > 8

Somma = ...
```


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
1	at
2-3	v0 – v1
4-7	a0 – a3
8-15	t0 – t7
16-23	s0 – s7
24-25	t8 – t9
26-27	k0 – k1
28	Gp
29	Sp
30	s8
31	Ra

MIPS Instruction Set: