Esercizio 1

Tradurre nel linguaggio assembler del MIPS la seguente funzione in linguaggio C, supponendo che i parametri di ingresso e il valore calcolato dalla funzione vengano passati sullo stack e che nell'implementazione assembler si possano usare solo i registri $s.

```
int  massimo(int x, int y, int z)
{
int max=x;
  if (y>max) max=y;
  if (z>max) max=z;
return max;
}
```

Esercizio 2

Tradurre nel linguaggio assembler del MIPS il seguente frammento di codice in linguaggio C, supponendo che il parametro di ingresso e il valore calcolato dalla funzione vengano passati tramite i registri Sa0 e $v0, rispettivamente.

```
  int a, val;
  …
  printf("dammi il valore di a:\n");
  scanf("%d",&a);
  val=fun(a)
  printf("il massimo e': %d",val);
  …
```

Esercizio 3

Completare il seguente frammento di codice assembler relativo all'implementazione di una procedura, supponendo che i registri $a0 e $a1 contengano i valori da passare alla funzione e i registri $v0 e $v1 i valori calcolati dalla funzione, inserendo le istruzioni necessarie laddove indicato

```
        …
        lw $a0, 4($s0)
        lw $a1, 0($s0)
        …                       # inserire qui le istruzioni necessarie
        jal fun
        …                       # inserire qui le istruzioni necessarie
        add $a0, $a0, $v0
        add $a1, $a1, $v1
        …

fun:    add $v0, $zero, $zero
        add $v1, $zero, $zero
        …                       # inserire qui le istruzioni necessarie
```

```
add $s0, $a0, $a1
sub $s1, $a0, $a1
add $v0, $s0, $v0
add $v1, $s1, $v1
…                          # inserire qui le istruzioni necessarie
jr $ra
```

Esercizio 4

Completare il seguente frammento di codice assembler, supponendo che i registri $s0 e $s1 contengano prima della chiamata i valori da passare alla funzione tramite lo stack e dopo la chiamata i valori calcolati dalla funzione letti dallo stack, inserendo le istruzioni necessarie laddove indicato.

```
…
la $t0, a
la $t1, b
lw $s0,0($t0)          # primo parametro da passare
lw $s1,0($t1)          # secondo parametro da passare
…                      # inserire qui le istruzioni necessarie
jal fun
…                      # inserire qui le istruzioni necessarie
sw $s0,0($t0)          # primo valore calcolato dalla funzione
sw $s1,0($t1)          # secondo valore calcolato dalla funzione
…
```