

```
1 # 1. |
2
3 .text
4     lw $s0, A      # Load the integers
5     lw $s1, B
6     lw $s2, C
7     lw $s3, D
8
9     sub $t0, $s0, $s1
10    sub $t1, $s2, $s3
11    add $t2, $t0, $t1
12    # works. should store (a - b) + (c - d) in Z ((1 - 5) + (2 - 4) = -4 + (-2) = -6)
13
14    sw $t2, Z
15
16 .data
17 A: .word 1
18 B: .word 5
19 C: .word 2
20 D: .word 4
21 Z: .word 0
```

```
1 # 2. |
2
3 .text
4     lw $s0, f      # Load the integers
5     lw $s1, g
6     lw $s2, h
7     lw $s3, i
8     lw $s4, j
9     la $s6, A      # Load the arrays
10    la $s7, B
11
12    sll $t0, $s3, 2    # t0 == i * 4
13    add $t0, $t0, $s6    # t0 += A
14    lw $t0, 0($t0)
15    sll $t1, $s3, 2    # t1 == i * 4
16    add $t1, $t1, $s7    # t1 += B
17    lw $t1, 0($t1)
18    add $t0, $t0, $t1    # t0 += t1
19    li $s0, 0
20    add $s0, $t0, $s2    # s0 == t0 + h ()
21    # works. should store A[i] + B[i] + h in f ($s0 = 3 + 4 + 69 = 76 = 0x4c)
22
23    sll $t2, $s4, 2
24    add $t2, $t2, $s6
25    lw $t2, 0($t2)
26    sll $t3, $s4, 2
27    add $t3, $t3, $s7
28    lw $t3, 0($t3)
29    sub $t2, $t2, $t3
30    li $s0, 0
31    add $s0, $t2, $s1
32    # works. should store A[j] - B[j] + g in f ($s0 = 5 - 6 + 42 = 41 = 0x29)
33
34 .data
35 A: .word 1,3,5,7,9
36 B: .word 2,4,6,8,10
37 f: .word 0
38 g: .word 42
39 h: .word 69
40 i: .word 1
41 j: .word 2
```

```
1 # 3.
2
3 .text
4     lw $s0, var1    # Load the integers
5     lw $s1, var2
6
7     li $v0, 1
8     mul $a0, $s0, $s1
9     syscall
10
11 .data
12 var1: .word 4
13 var2: .word 6
```

```
1 # 4.
2
3 .text
4     lw $s0, var1    # Load the integers
5     lw $s1, var2
6
7     li $v0, 1        # set the syscall service number to print_integer
8     div $a0, $s0, $s1 # store the multiple of the two numbers input in $a0, used for printing
9     syscall          # make the syscall
10
11 .data
12 var1: .word 20
13 var2: .word 5
```