1. Exercise 2.31

sll $s2, $a2, 2 # max i = 2500 * 4
sll $s3, $a3, 2 # max j = 2500 * 4
add $v0, $zero, $zero # $v0 = 0
add $t0, $zero, $zero # i = 0
outer: add $t4, $a0, $t0 # $t4 = address of array 1[i]
lw $t4, 0($t4) # $t4 = array 1[i]
add $t1, $zero, $zero # j = 0
inner: add $t3, $a1, $t1 # $t3 = address of array 2[j]
lw $t3, 0($t3) # $t3 = array 2[j]
bne $t3, $t4, skip # if (array 1[i] != array 2[j]) skip $v0++
addi $v0, $v0, 1 # $v0++
skip: addi $t1, $t1, 4 # j++
bne $t1, $a3, inner # loop if j != 2500 * 4
addi $t0, $t0, 4 # i++
bne $t0, $a2, outer # loop if i != 2500 * 4

The code determines the number of matching elements between the two arrays and returns this number in register $v0.

Ignoring the four instructions before the loops, we see that the outer loop (which iterates 2500 times) has three instructions before the inner loop and two after. The cycles needed to execute these are $1 + 2 + 1 = 4$ cycles and $1 + 2 = 3$ cycles, for a total of 7 cycles per iteration, or $2500 \times 7$ cycles.
The inner loop requires $1 + 2 + 2 + 1 + 1 + 2 = 9$ cycles per iteration and it repeats $2500 \times 2500$ times, for a total of $9 \times 2500 \times 2500$ cycles.

The total number of cycles executed is therefore $(2500 \times 7) + (9 \times 2500 \times 2500) = 56,267,500$. The overall execution time is therefore $(56,267,500) / (2 \times 10^9) = 28$ ms.
Note that the execution time for the inner loop is really the only code of significance.

2. Exercise 2.47

To test for loop termination, the constant 401 is needed. Assume that it is placed in memory when the program is loaded:

lw $t8, AddressConstant401($zero) # $t8 = 401
lw $t7, 4($a0) # $t7 = length of a[]
lw $t6, 4($a1) # $t6 = length of b[]
add $t0, $zero, $zero # initialize i = 0

Loop: slt $t4, $t0, $zero # $t4 = 1 if i < 0
bne $t4, $zero, IndexOutOfBounds # if i < 0, goto Error
slt $t4, $t0, $t6 # $t4 = 0 if i >= length ($t6 is length of b[])
beq $t4, $zero, IndexOutOfBounds # if i >= length, goto Error
slt $t4, $t0, $t7 # $t4 = 0 if i >= length ($t7 is length of a[])
beq $t4, $zero, IndexOutOfBounds # if i >= length, goto Error
add $t1, $a1, $t0 # $t1 = address of b[i]
lw $t2, 8($t1) # $t2 = b[i]
add $t2, $t2, $s0 # $t2 = b[i] + c
add $t3, $a0, $t0 # $t3 = address of a[i]
sw $t2, 8($t3) # a[i] = b[i] + c
addi $t0, $t0, 4 # i = i + 4
slt $t4, $t0, $t8 # $t4 = 1 if $t0 < 401, i.e., i <= 100
bne $t4, $zero, Loop # goto Loop if i <= 100

There are 4 instructions outside the loop, and 14 instructions inside the loop. The number of
instructions executed is 4 + 101 x 14 = 1418.
The number of data references made (i.e. lw and sw) is 3 + 101 x 2 = 205.
Note: the solution is not unique.

3. Exercise 2.49

From Figure 2.48, 36% of all instructions for SPEC2000int are data access instructions. Thus, for
every 100 instructions there are 36 data accesses, yielding a total of 136 memory accesses (1 to read
each instruction x 100 and 36 to access data).

a. The percentage of all memory accesses that are for data = 36/136 = 26.47%.

b. Assuming two-thirds of data transfers are loads, the percentage of all memory accesses that
are reads = (100 + (36 x 2/3))/136 = 91.18%.