**Program 1: add.asm**

.data

x: .word 0

y: .word 0

z: .word 0

nl: .asciiz "\n"

.text

main:

li $v0, 5 # Read x

syscall

la $t0, x

sw $v0, 0($t0)

li $v0, 5 # Read y

syscall

la $t0, y

sw $v0, 0($t0)

la $t0, x # z = x + y

lw $t1, 0($t0)

lw $t2, 4($t0)

add $t3, $t1, $t2

la $t0, z

sw $t3, 0($t0)

li $v0, 1 # Print z

lw $a0, 0($t0)

syscall

li $v0, 4

la $a0, nl

syscall

la $t0, x

lw $t1, 0($t0)

lw $t2, 4($t0)

sub $t3, $t1, $t2

sw $t3, 8($t0)

li $v0, 1

lw $a0, 8($t0)

syscall

li $v0, 4

la $a0, nl

syscall

li $v0, 10 # Exit

syscall

**In the data section**

1. x = 0
2. y = 0
3. z = 0
4. nl = “\n” (newline character)

**Instructions**

1. Read x from input ($v0 = 5)
2. Make the system call to read from input ($v0 = input value of x)
3. $t0 now points to the memory address of x
4. $t0 = $v0 (0($t0) contains input value of x)
5. Read y from input ($v0 = 5)
6. Make the system call to read from input ($v0 = input value of y)
7. $t0 points to the memory address of y
8. $t0 = $v0 ($v0 contains input value of y)
9. $t0 points to the address of x
10. $t1 = $t0 (Read 1st value stored in $t0)
11. $t2 = $t0 + 4 bytes (Get x and load into $t0)
12. $t3 = $t1 + $t2 (z = x + y)
13. $t0 now points to the memory address of z
14. $t0 = $t3 (0$t0 contains z)
15. Print integer to console ($v0 = 1)
16. Load word from location pointed by $t0 (z) into $a0
17. Make system call to print z to console
18. Print string to console ($v0 = 4)
19. Load newline into $a0
20. Make system call to print newline to console
21. Load value of x into $t0
22. Load memory address of x into $t1
23. Load memory address of y into $t2
24. $t3 = $t1 - $t2 (z = x - y)
25. $t3 -> $t0 + 8 bytes (Save value of $t3 (z) into the 3rd offset of $t0)
26. Print integer to console ($v0 = 1)
27. $a0 = $t0 + 8 bytes (Get z and load into $t0)
28. Make system call to print the value of z to console
29. Print string to console ($v0 = 4)
30. Load newline into $a0
31. Make system call to print newline to console
32. Prepare call to exit ($v0 = 10)
33. Exit the program

**Program 2: countdown.asm**

.data

cstart:

.word 10

nl:

.asciiz "\n"

.text

main:

la $t0, cstart # step 1: Load counter

lw $s0, 0($t0)

loop:

li $v0, 1 # step 2: Print counter

or $a0, $s0, $zero

syscall

li $v0, 4 # Print newline

la $a0, nl

syscall

bne $s0, $zero, continue # If counter != 0, go to continue

li $v0, 10 # exit

syscall

continue:

addi $s0, $s0, -1 # step 4: decrement counter

b loop # step 5: go to 2

**In the data section**

1. cstart is a looping variable initialized with value of 10 (cstart=10)
2. nl = “\n” (newline character)

**Instructions**

1. $t0 = cstart (Load cstart into register $t0)
2. Make the system call to read from input ($v0 = input value of x)
3. $s0 = 10 (load word from 0$t0 into $s0)

**The loop:**

# 1st iteration:

1. Print integer to console ($v0 = 1)
2. Copy word from $s0 into $a0 ($a0 = 10) and make the syscall to print 10 to console
3. Print string to console ($v0 = 4)
4. Load address to $a0 ($a0 = “\n”) and make the syscall to print newline to console
5. Check if $s0 (the counter) is not equal to 0 and if true decrement the counter by 1 (addi $s0, $s0, -1) and continue the loop

# 2nd iteration:

1. Print integer to console ($v0 = 1)
2. Copy word from $s0 into $a0 ($a0 = 9) and make the syscall to print 9 to console
3. Print string to console ($v0 = 4)
4. Load address to $a0 ($a0 = “\n”) and make the syscall to print newline to console
5. Check if $s0 (the counter) is not equal to 0 and if true decrement the counter by 1 (addi $s0, $s0, -1) and continue the loop

…

# Last iteration:

1. Print integer to console ($v0 = 1)
2. Copy word from $s0 into $a0 ($a0 = 0) and make the syscall to print 0 to console
3. Print string to console ($v0 = 4)
4. Load address to $a0 ($a0 = “\n”) and make the syscall to print newline to console
5. Check if $s0 (the counter) is not equal to 0. This is now false.   
     
     
   **Finally**
6. Send code 10 to register $v0 (exit the program)
7. Make the syscall to exit