

Solutions to Lab1 Intro. to SPIM

Psedo- Instruction	Instruction	Opcode	rs	rt	Immediate data or Offset rd, shamnt, func
-----------------------	-------------	--------	----	----	--

li \$8,0x3210

34083210	ori \$8, \$0, 0x3210	001101	00000	01000	0011 0010 0001 0000
----------	----------------------	--------	-------	-------	---------------------

li \$09,0x76543210

3c017654	lui \$1, 0x7654	001111	00000	00001	0111 0110 0101 0100
34293210	ori \$9, \$1, 0x3210	001101	00001	01001	0011 0010 0001 0000

sge \$11, \$8, \$9

15280003	bne \$9, \$8, 12	000101	01001	01000	0000 0000 0000 0011
340b0001	ori \$11, \$0, 1	001101	00000	01011	0000 0000 0000 0001
10000003	beq \$0, \$0, 12	000100	00000	00000	0000 0000 0000 0011
00000000	nop	000000	00000	00000	0000 0000 0000 0000
0128582a	slt \$11, \$9, \$8	000000	01001	01000	01011 00000 101010

mul \$12, \$11, \$10

716a6002	mul \$12, \$11, \$10	011100	01011	01010	01100 00000 000010
----------	----------------------	--------	-------	-------	--------------------

bge \$11, \$0, infloop

0160082a	slt \$1, \$11, \$0	000000	01011	00000	00001 00000 101010
10200000	beq \$1, \$0, 0	000100	00001	00000	0000 0000 0000 0000

Part 2	Psedo- Instruction	Instruction	Opcode	rs	rt	Immediate data or Offset rd, shamnt, func
--------	-----------------------	-------------	--------	----	----	--

340a0019	li \$t2, 25	ori \$10, \$0, 25	001101	00000	01010	0000 0000 0001 1001
3c011001	lw \$t3, value	lui \$1, 4097	001111	00000	00001	0001 0000 0000 0001
8c2b0000		lw \$11, 0(\$1)	100011	00001	01011	0000 0000 0000 0000
014b6020	add \$t4, \$t2, \$t3	add \$12, \$10, \$11	000000	01010	01011	01100 00000 100000
014b6822	sub \$t5, \$t2, \$t3	sub \$13, \$10, \$11	000000	01010	01011	01101 00000 100010
3c011001	sw \$t5, Z	lui \$1, 4097	001111	00000	00001	0001 0000 0000 0001
ac2d0004		sw \$13, 4(\$1)	101011	00001	01101	0000 0000 0000 0100
3402000a	li \$v0, 10	ori \$2, \$0, 10	001101	00000	00010	0000 0000 0000 1010

```

# Solutions to Lab2.1 Calculate 1 + 2 + 3 + ... + N
.data
prompt: .asciiz "\n Please input a value for N = "
result: .asciiz "The sum of the integers 1 to N = "
bye:    .asciiz "\n Good-bye!"
.globl main
.text

main:
    li $v0, 4 # System call code for Print String
    la $a0, prompt # load address of prompt into $a0
    syscall # Print the prompt message
    li $v0, 5 # System call code for Read Integer
    syscall # Read N into $v0
    blez $v0, end # branch to end if v0 <= 0
    li $t0, 0

loop:
    add $t0, $t0, $v0 # Sum of integers in $t0
    addi $v0, $v0, -1 # Decrement N
    bnez $v0, loop # branch to loop if v !=0
    li $v0, 4 # System call code for Print String
    la $a0, result # load address of message into $a0
    syscall # print the string
    li $v0, 1 # System call code for Print Integer
    move $a0, $t0 # move value to be printed to $a0
    syscall # print sum of integers
    b main # branch to main

end:
    li $v0, 4 # System call code for Print String
    la $a0, bye # load address of message into $a0
    syscall # print the string
    li $v0, 10 # System call code for terminate
    syscall # return control to system

```

```

# Solutions to Lab2.2 Sum of N intergers, a[0] + a[1] +
... a[N-1]
        .data
n:        .word 10          # n = 10
# array of 10 intergers
a:        .word 9, 7, 8, 6, 4, 5, 2, 10, 1, 3
        .globl main
        .text
main:
        and $s0, $0, $0 # $s0 (sum) = 0
        and $t0, $0, $0 # $t0 (i) = 0
        la $t4, n        # address of n
        lw $t4, 0($t4)    # t4 = n
        la $t2, a         # $t2 = address of a[i], i = 0
loop2:
        lw $t3, 0($t2) # load a[i]
        add $s0, $s0, $t3 # increment sum
        addi $t0, $t0, 1 # increment i
        addi $t2, $t2, 4 # increment address of a[i]
        slt $t5, $t0, $t4 # is $t0 < $t4 ?
        bne $t5, $0, loop2 # branch if so
# Output Sum
        li $v0, 1 # Load 1=print_int into $v0
        add $a0, $s0, $zero # Load first number into $a0
        syscall # Output the prompt via syscall
        li $v0, 10 # System call code for terminate
        syscall # return control to system

```