```
CS 2734, Org II, Fall 2001
First Exam, Selected Answers
================================================================================
1.  (a) 58 (base 10) = 111010 (base 2)
       -58 (base 10 = ~111010 + 1 = 1111 1111 1100 0101 + 1 =
                                    1111 1111 1100 0110
    (b) 0100 0000 0011 0100  (48 more 0's)  (binary) =

        0 10000000011  0100  (48 more 0's)  (broken into fields) =

        (-1)^Sign x 2^(Exponent - Bias) x (1 + Significand) =
        (-1)^0    x 2^(1027     - 1023) x (1 + 0.01 (base 2)) =
          1       x 2^4                 x 1.25 = 16 x 1.25 = 20.0
================================================================================
2 and 3.
    # CS 2734, Computer Organization II, Fall 2000
    # MIPS program giving answer to Exam1, question 2
            .globl main
    main:       addu    $s7, $zero, $ra
    ########## Start of answer to Question 2 ######################
            .data                                                  #
    A:      .word 2, 3, 5, 7, 11, 13, 17, 19, 23, 29               #
            .text                                                  #
            la      $s0, A          # address of A                 #
            addi    $t0, $0, 0      # loop counter, start at 0     #
            addi    $t1, $0, 10     # to terminate loop            #
            add     $t2, $0, $s0    # address of item in A         #
            addi    $s1, $0, 0      # running sum                  #
    Loop:   lw      $t3, 0($t2)     # store current $t0 into A     #
            addi    $t0, $t0, 1     # increment loop counter       #
            addi    $t2, $t2, 4     # increment pointer into A     #
            add     $s1, $s1, $t3   # add A[i] into running sum    #
            bne     $t0, $t1, Loop  # branch back to form loop     #
    ########## End of answer to Question 2 ########################

        ## Print the sum just calculated (129)
            li      $v0, 1
            add     $a0, $s1, $0
            syscall
            jal     Newl

    ########## Start of answer to Question 3 (first part) #########
            add     $a0, $s0, $0    # addr of A in $a0             #
            jal     Avals           # call Avals                   #
    ########## End of answer to Question 3 (first part) ###########

            add     $a0, $v0, $0    # print $v0 just returned (5)
            li      $v0, 1
            syscall
            jal     Newl
    ########## Finish main###################
            addu    $ra, $zero, $s7
```

```
            jr      $ra

    ########## Start of answer to Question 3 (second part) ########
    Avals:                                                        #
            addi    $sp, $sp, -4    # room for $ra on stack       #
            sw      $ra, 0($sp)     # save $ra                    #
            lw      $t4, 0($a0)     # $t4 = A[0]                  #
            lw      $t5, 4($a0)     # $t5 = A[1]                  #
            add     $v0, $t4, $t5   # $v0 = A[0] + A[1]           #
            lw      $ra, 0($sp)     # restore $ra from stack      #
            jr      $ra             # return                      #
    ########## End of answer to Question 3 (second part) ##########

    ########## write newline ######################################
    Newl:   li      $v0, 4
            la      $a0, Newline
            syscall
            jr      $ra
            .data
    Blank:      .asciiz  " "
    ############### output ##############################
    # four06% spim -file exam1.s
    # 129
    # 5
    ######################################################
```

==========

4.

Addr	Instruction	op	rs	rt	rd	sh	ft	
32	0x2008000a	001000	00000	01000	00000	00000	001010	
		8	0	8			10	addi $t0, $0, 10
36	0x01284820	000000	01001	01000	01001	00000	100000	
		0	9	8	9		32	add $t1, $t1, $t0
40	0x1509fffe	000101	01000	01001	11111	11111	111110	
		5	8	9			-2	bne $t0, $t1, -2

==========

5.

(a) D = (A and (not B) and (not C)) or ((not A) and (not B) and C)
or ((not A) and B and (not C)).

(b) Standard construction.

==========