2312
Spring 2018 Key 2

1. f = (g − 7) + (h + 79); f=$s0, g=$t0, h=$t1

a) addi $t2, $t0, -7
addi $t3, $t1, 79
add $s0, $t2, $t3

6 pts

Correct solutions must have 2 addi since there are 2 constants and no subi command. must have one add command for register operation.

b) R-type | opcode | rs | rt | rd | shamt | funct |
31 26 25 21 20 16 15 11 10 6 5 0

I-type | opcode | rs | rt | immediate |
31 26 25 21 20 16 15 0

addi $t2, $t0, -7 | 8hex | 8 | 10 | -7 |
opcode rt rs Immed 001000 01000 01010 1111 1111 1111 1001

addi $t3, $t1, 79 | 8hex | 9 | 11 | 79 |
001000 01001 01011 0000 0000 0100 1111

add $s0, $t2, $t3 | 0 | 10 | 11 | 16 | 0 | 20 hex |
000000 01010 01011 10000 00000 100000

00100001 0000 1010 1111 1111 1111 1001
00100001 0000 1010 0000 0000 0100 1111
00000001 0100 1011 1000 0000 0001 0100

9 pts

Correct solutions must have: A) Two instructions with 001000 as opcode and B) one instruction with 000000 in opcode and 100100 in function.

2. $t0 = 00FF00FF = 0000 0000 1111 1111 0000 0000 1111 1111
$t1 = ABCD1234 = 1010 1011 1100 1101 0001 0010 0011 0100

a) sll $t2, $t0, 8 # shift $t0 8 left.
$t2 = 0111 1111 1000 0000 0111 1111 1000 0000
or $t2, $t2, $t1 # or $t2 and $t1
$t2 = 0111 1111 1000 0000 0111 1111 1000 0000
or t1 = 1010 1011 1100 1101 0001 0010 0011 0100
8pts $t2 = 1111 1111 1100 1101 0111 1111 1011 0100
or FFCD 7FB4

b) sll $t2, $t0, 3 # shift $t0 3 left
$t2 = 0000 0111 1111 1000 0000 0111 1111 1000
andi $t2, $t2, -9 # and $t2 with -9
$t2 = 0000 0111 1111 1000 0000 0111 1111 1000
andi -9 = 1111 1111 1111 1111 1111 1111 1111 0111
8pts $t2 = 0000 0111 1111 1000 0000 0111 1111 0000
or 07F807F0

c) srl $t2, $t0, 6 # shift $t0 right 6
$t2 = 0000 0000 0000 0011 1111 1100 0000 0011
andi $t2, $t2, 0x0000EEA0 # and $t2 with 0x0000EEA0
$t2 = 0000 0000 0000 0011 1111 1100 0000 0011
andi const = 0000 0000 0000 0000 1110 1110 1010 1101
8pts $t2 = 0000 0000 0000 0000 1110 1100 0000 0001
or 0000EC01

3. Since there are 3 registers, we have input parameters, so it must be a function. Return must be an int since problem states every thing ~~must be~~ is an integer.

10 pts

```
int funC int x, int y, int z)
{        x+y-z;
  return x+y+z;
}
```

Correct solutions must have int return type
x, y, z input parameters
calculate a sum and return

4(a)

iteration	0	1	2	3	4	5	6	7	8	9
$t2	-	1	1	1	1	1	1	1	1	0
$t1	8	7	6	5	4	3	2	1	0	3
$s2	0	3	6	9	12	15	18	21	24	3

6 pts

$s2 = 30 ?

b) while (i > 0)
{ i--;
  $t = 3;
}

Correct solutions must have loop control stat

5 pts

c) ~~5 iteration~~.
5 instructions per iteration of the loop.
termination takes 2 instructions
5N+2

5 pts

2312
Spring 2018 Key 2 Pg of 5

```
S  add $t0, $zero, $zero   # i = 0
Loop 1: beq $t0, $s0, Done1   # break if i = a
        add $t1, $zero, $zero  # j = 0
Loop 2: beq $t1, $s1, Done2   # break if j = b
        add $t2, $t0, $t1      # $t2 = i + j
        sll $t3, $t1, 3        # $t3 = j x 2^3 = j x 8 [2 for j x 2, 4 for offset]
        add $t4, $s2, $t3      # $t4 = address of D[j x 2]
        sw $t2, 0($t4)         # store $t2 at D[j x 2]
        addi $t1, $t1, 1       # j++
        j Loop2                # Go back to j loop condition
Done2:  addi $t0, $t0, 1       # i++
        j Loop1                # Go back to j loop condition
Done1:
```

Correct Solution Must have
- 2 "loop" labels
- 2 "done" labels
- 2 termination statements (bne or beq)
- 2 jumps to labels
- 2 addi for $t0 and $t1 for increment
- mult or sll for j*2 and offset
- sw statement

20 pts

2312
Spring 2018 Key 2

15 pts

6. andi $t0, $s0, 128 # 128 is 8th bit, so compare only to that. all other bits will be 0
beqi $t0, $zero, ELSE # if it is all zeros, so not 1
addi: $s1, $s1, 2 ~~# this operates if it~~
this runs only if it is 1. so we add two
j: Done # skip our Else statement
ELSE: ori $s0, $s0, 128 # or the number with a 128 to set that bit to 1
DONE: # label to skip the else

Correct Solution must have
- and i statement or shift ~~left~~ right + 7
- Evaluate if the result is zero
- beq or bne with an else statement
- j to Done
- setting 8th bit to 1 ether ori or addi