Drew Pulliam : DTP180002
3340.004

UTD

CS/SE 3340 - Assignment#2
Due Date: 3/4/20, 11:59 pm

1-For the initial register values shown below, what is the value of $t0, $t1 and $t2 after executing each instruction in binary system? What is the final value of $t0, $t1 and $t2 in hexadecimal system?

$t0 = 0xAAAAAAAA (hex) = 1010 1010 1010 1010 1010 1010 1010 1010
$t1 = 0x87654321 (hex) =   1000 0111 0110 0101 0100 0011 0010 0001

```
                                              final values
sll  $t2, $t0, 4    $t2 = 0x aaaa aaa0    $t0 = 0x 0000 0001
or   $t2, $t2, $t1  $t2 = 0x afef ebal    $t1 = 0x 8765 4321
nor  $t2, $t2, $t1  $t2 = 0x 5010 145e    $t2 = 0x 0000 0007
slt  $t0, $t2, $t1  $t0 = 0x 0000 0000
sltu $t0, $t2, $t1  $t0 = 0x 0000 0001
srl  $t2, $t0, 3    $t2 = 0x 0000 0000
addi $t2, $t2, 7    $t2 = 0x 0000 0007
```

2-For the memory locations in the table below, write MIPS code to add all the elements, placing the result in the smallest memory location. Use a minimum number of MIPS instructions. Assume the base address of Array is stored in register $s0.

```
          la $t0, 60($s0)
          la $t1, 36($s0)
loop: beq $t0, $t1, end
          lw $t3, 0($t0)
          add $t2, $t2, $t3
          addi $t0, $t0, -4
          j loop
end:  sw $t3, 0($t1)
```

Word Address	Data
15	4
14	5
13	6
12	8
11	1
10	3

3-Assume we place the following MIPS code starting at location 8000 in memory

```
LOOP:   slt   $t2, $0,  $t1
        beq   $t2, $0,  DONE
        addi  $t1, $t1, -1
        addi  $s2, $s2, 2
        j     LOOP
DONE:
```

#3| slt $t2, $0, $t1   — R-format

0x0009 502a

| 0000 | 0000 | 0000 1001 | 0101 0000 | 00 | 10 1010 |

decimal

| 0 | 0 | 9 | 10 | 0 | 42 |

beq $t2, $0, DONE   — I-format

0x1140 0003

| 0001 00 | 01 0100 | 0000 0000 | 0000 0000 | 0000 0011 |

decimal

| 4 | 10 | 0 | 3 |

addi $t1, $t1, -1   — I-format

0x 2129 ffff

| 0010 00 | 01 0010 | 1001 | 1111 1111 1111 1111 |

decimal

| 8 | 9 | 9 | -1 |

addi $s2, $s2, 2   — I-format

0x 2252 0002

| 0010 00 | 10 0101 | 0010 | 0000 0000 0000 0010 |

decimal

| 8 | 18 | 18 | 2 |

j LOOP   — J-format

0x 0800 07D0

| 0000 10 | 00 0000 0000 0000 0111 1101 0000 |

| 2 | 2000 |

© the only time the code modifies $s2 is addi $s2, $s2, 2

0 + 2 = 2

$s2 has a value of 2_{10}

ⓓ | $s1 = A
$s2 = B
$t1 = i
$t2 = temp

```
temp = (0 < i) ? 1 : 0;
while (temp != 0){
    i = i - 1;
    B = B + 2;
}
//done
```