CSE 3038 – QUIZ 2 – Spring 2022 A

Name: **Signature:**

PART A - 70 PTS (Each correct test answer worths 7 pts, whereas an incorrect answer has a penalty of 1.75 pts).

1) Provide the type and assembly language instruction for the following hex value: **0x022A8020**
A) add $s0, $s1, $t2
B) addi $s0, $s1, 2
C) sub $s0, $s1, $s2
D) sll $s0, $s1, 3
E) srl $s0, $s1, 4

2)Provide the hex value of following assembly language instruction: **lw $5, 8($11)**
A) 0x8CAB0008
B) 0x8DAB0004
C) 0x8D650008
D) 0xACAB0008
E) 0xAD650008

3) Assume the following register contents:
$s0 = 0xABCD1234 $s1 = 0xFFFFEEEE
For the register values shown above, what is the value of $t2 for the following sequence of instructions ?

sll $t2, $s0, 7
or $t2, $t2, $s1

A) 0xE6891C00
B) 0xE2D92780
C) 0xFFFFFFFF
D) 0xFFFEEEEE
E) 0xFFFFFEEE

4) There are several different instructions that can be used to change the control flow in a MIPS program. Which of the following one has the greatest range? (In other words, which instruction can be used to go the furthest away relative to where the instruction is located?)
A) j B) jr C)jal D) beq e)bne

5) For the following C statement, choose the minimal sequence of MIPS assembly instructions that does the identical operation. Assume $t0 = A and $s0 is the base address of C.

A = C[4] << 4

A)
```
lw $t3, 4($s0)
sll $t0, $t3, 4
```
B)
```
lw $t3, 16($s0)
sll $t0, $t3, 4
```
C)
```
lw $t3, 16($t0)
srl $t0, $t3, 8
```
D)
```
lw $t3, 16($s0)
sll $t0, $t3, 4
```
E)
```
sw $t3, 4($t0)
srl $t0, $t3, 16
```

6) What would be the value of register $t0 after execution of following assemble code ?

```
addi $t2, $zero, 6
add $t0, $zero, $zero
label : add $t3, $t2, $zero
        sll $t3, $t3, 2
        add $t0, $t0,$t3
        addi $t2,$t2,-1
        bne $t2,$zero,label
```

A)21 B)42 C)84 D)105 E)126

7) Consider the segment of assembly code below. To the left of each instruction, we write the memory address (the byte address given in hexadecimal) where each instruction is stored. Write the binary representation of bne instruction (with all fields).

Memory Address	Label	Instruction
0001A0	back:	add $s2, $s3, $s4
……..		…….
……..		……..
0001F8		bne $s1, $zero, back

A) - 92 B) - 22 C) - 21 D) - 23 E) - 88

8) Consider the following MIPS loop:

```
LOOP:  slt $t2, $zero, $t1
       beq $t2, $zero, DONE
       subi $t1, $t1, 1
       addi $s2, $s2, 4
       j LOOP
DONE:
```

Assume that the register `$t1` is initialized to the value 10. What is the value in register `$s2` assuming `$s2` is initially zero?

A) 5 B)30 C)10 D)40 E)20

9) Write the shortest MIPS program segment (with the minimum number of instructions) that implements the pseudo-instruction: `beven $a0, LOC`.

Note that this instruction causes a branch to LOC if the number in $a0 is even.

A) ori $t1, $a0, 1
bne $t1, $zero, LOC

B) andi $t1, $a0, 1
bne $t1, $zero, LOC

C) andi $t1, $a0, 0
beq $t1, $zero, LOC

D) ori $t1, $a0, 0
bne $t1, $zero, LOC

E) andi $t1, $a0, 1
beq $t1, $zero, LOC

10) Assume that "number" is an 32-bit immediate operand that is equal to :

0000 0000 0010 0101 0000 0000 0011 0110

Which of the followings is asequence of actual MIPS instructions to accomplish the following:

if ($s3 != number) goto Label

a) add $t0, $zero, $zero
lui $t0, 33
addi $t0, $t0, 50
beq $s3, $t0, Label

b) add $t0, $zero, $zero
lui $t0, 50
addi $t0, $t0, 33
bne $s3, $t0, Label

c) add $t0, $zero, $zero
lui $t0, 37
addi $t0, $t0, 54
bne $s3, $t0, Label

d) add $t0, $zero, $zero
lui $t0, 54
addi $t0, $t0, 37
bne $s3, $t0, Label

e) add $t0, $zero, $zero
lui $t0, 21
addi $t0, $t0, 38
bne $s3, $t0, Label

PART B - 30 PTS

Assume that "MyArray" is an integer array where the base address is in **`$s0`**. The register **`$s1`** has the length of the array (i.e., the number of elements in the array). Translate the following loop into C.

Note: bgt (Branch on Greater Than) is a pseudo-instruction.
`bgt src1,src2,label:` Conditionally branch to the instruction at the label if the contents of register src1 are greater than src2

```
mysterious:    add $t0, $zero, $s0
               addi $t1, $s1, -1
               sll $t1, $t1, 2
               add $t1, $s0, $t1
     loop:     lw $t2,0($t0)
               lw $t3,0($t1)
               sw $t2, 0($t1)
               sw $t3, 0($t0)
               addi $t1, $t1, -4
               addi $t0, $t0, 4
               bgt $t1, $t0, loop
```

SOLUTION

Assume that "MyArray" is an integer array where the base address is in **$s0**. The register **$s1** has the length of the array (i.e., the number of elements in the array). Translate the following loop into C.

```
mysterious: add $t0, $zero, $s0          #$t0 = &MyArray[0]
            addi $t1, $s1, -1            #$t1 = length – 1
            sll $t1, $t1, 2              #$t1 = 4 x (length – 1)
            add $t1, $s0, $t1            #$t1 = Addr. of last array element
      loop: lw $t2,0($t0)                #$t2 = MyArray[0]
            lw $t3,0($t1)                #$t3 = MyArray[length - 1]
            sw $t2, 0($t1)                     #Swap $t2 & $t3
            sw $t3, 0($t0)
            addi $t1, $t1, -4            #go downwards
            addi $t0, $t0, 4             #go upwards
            bgt $t1, $t0, loop           #crosspoint
```

```
i = 0;
j = length – 1;
while (i < j) {
      temp = MyArray[i];
      MyArray[i] = MyArray[j];
      MyArray[j] = temp;
      i++;
      j--;
}
```