Assignment 2

1. Assume that $a0 contains an input variable and is initialized to an integer n. Assume the register $v0 stores the output.

Add an appropriate comment to each line of code given. What does this program do?

addi $t0, $zero, 0 : **adds 0 and $zero and stores into $t0.**

addi $t1, $zero, 2 : **adds 2 and $zero and stores in $t1**

loop: slt $t2, $t1, $a0 : **if $t1 < $a0, then 1 into $t2**

beq $t2, $zero, done : **if $t2 = $zero, then jump to done**

add $t0, $t0, $t1 : **add $t0 and $t1 and store into $t0**

addi $t1, $t1, 2 : **add $t1 and 2 and store into $t1**

j loop : **jump to loop**

done: add $v0, $t0, $zero: **adds $t0 and $zero and stores in $v0**

2. Suppose you have an array beginning at an address stored in $t0. Its contents are 0,1,2,3,4,5,6,7,8,9. After the following instructions, what will be the values in this array?

lw $t1, 4($t0)

**load word/array of $t0, add 4 and store into $t1**

sll $t1, $t1, 3

**shift logical left of the contents in $t1 by 3 and store into $t1**

add $t0, $t0, $t1

**add $t0 and $t1 and store in $t0**

sw $t1, 0($t0)

**store $t0 into the memory of $t1**

**3456789000**

3. Use the MIPS register table (Fig. A.6.1) and opcode map (Fig. A.10.2) to convert the instructions below to 32b MIPS instruction object code (in hexadecimal format). (6)

addi $t0, $zero, 0 0x**20090000**

addi $t1, $zero, 2 0x**200A0000**

4. Disassemble the following MIPS object code into source code instructions. Use register names, such as $t2, instead of numbers, such as $20. (8)

a. 0x00069980

**0000 0000 0000 0110 1001 1001 1000 0000**

**000000 00000 00110 10011 00110 000000**

**0 0 6 19 6 0**

**$zero $zero $a2 $s3 $a2 $zero**

**sll $s3 $a2, 6**

b. 0x03C00011

**0000 0011 1100 0000 0000 0000 0001 0001**

**000000 11110 00000 00000 00000 010001**

**0 30 0 0 0 17**

**$zero $fp $zero $zero $zero $s1**

**sub $zero $fp, $zero**

c. 0x250C0014

**0010 0101 0000 1100 0000 0000 0001 0100**

**00100 01000 01100 00000 00000 010100**

**4 8 12 0 0 20**

**$a0 $t0 $t4 $zero $zero $s4**

d. 0x01936824

**0000 0001 1001 0010 0110 1000 0010 0100**

**000000 01100 10010 01101 00000 100100**

**0 12 18 13 0 18**

**$zero $t4 $s2 $t5 $zero $s2**

5. Generate the MIPS assembly code for the given C-code below. Assume that a is stored in $t0, b is stored in $t1, c is stored in $s0, the address of arrayA is stored in $s6 and the address of arrayB is stored in $s7. (12)

arrayA[5] = a + b + c;

c = c - arrayA[arrayB[2]];

**la $t0, a**

**la $t1, b**

**la $s0, c**

**lw $t3, 20($s6)**

**add $t0, $t1, $s0**

**add $t3, $t0, $t3**

**lw $t3, 8($s7)**

**sub $s0, $s0, $t3**

**sw $s0, 28($t3)**