1. Assume that registers $s0 and $s1 hold the values 0x80000000 and 0xD0000000 , respectively.

a. What is the value of $t0 for the assembly code: add $t0, $s0, $s1

0x80000000+0xD0000000 = 150000000(hex)

b. Is the result in $t0 the desired result, or has there been overflow?

overflow

c. For the contents of registers $s0 and $s1 as specified above, what is the value of $t0 for the following assembly code?
sub $t0, $s0, $s1

0x80000000 - 0xD0000000 = -50000000(hex)

d. Is the result in $t0 the desired result, or has there been overflow?

No overflow

e. For the contents of registers $s0 and $s1 as specified above, what is the value of $t0 for the following assembly code?
add $t0, $s0, $s1
add $t0, $t0, $s0

(0x80000000*2) + 0xD0000000 = 1d0000000(hex)

f. Is the result in $t0 the desired result, or has there been overflow?

overflow

2. Assume that $s0 holds the value 128.
a. For the instruction add $t0 , $s0 , $s1 , what is the range(s) of values for $s1 that would result in overflow?

2^{31} – 128 to (2^{31} - 1)

b. For the instruction sub $t0 , $s0 , $s1 , what is the range(s) of values for $s1 that would result in overflow?

-2^{31} to $-(2^{31} – 128)$

c. For the instruction sub $t0 , $s1 , $s0 , what is the range(s) of values for $s1 that would result in overflow?

-2^{31} to $-(2^{31} – 127)$

3.Write the MIPS assembly instruction and its binary encoding to accomplish this task: If the contents of R3 and R4 are equal, skip the next 10 instructions and execute the 11th.

Instruction:	Binary equivalent
beq R3, R4, 0x28	00010000011001000000000000101000

4. What the final effect of the MIPS instruction 0x3C000088?

Nothing

5. Assume $t0 holds the value 0x00101000 . What is the value of $t2 after the following instructions?

```
slt $t2, $0, $t0
bne $t2, $0, ELSE
j DONE
ELSE:
addi $t2, $t2, 2
DONE:
```

Value of $t2 after the code completes: 3

6. The following instruction is not included in the MIPS instruction set:

rpt $t2, loop # if(R[rs]>0) R[rs]=R[rs]-1, PC=PC+4+BranchAddr

a. If this instruction were to be implemented in the MIPS instruction set, what is the most appropriate instruction format?

I-type

b. What is the shortest sequence of MIPS instructions that performs the same operation?

```
loop:
      ble  $t2, $0, done   #if(R[rs]<=0) goto done

       addi $t2, $t2, -1    #R[rs]=R[rs]+(-1)

      j loop               #goto loop PC=PC+4+BranchAddr

      done:
```

7. Consider the MIPS loop on the right.

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

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

20

b. Write the equivalent C code routine. Assume that the registers $s1 , $s2 , $t1 , and $t2 are integers A , B , i , and temp, respectively.

```
 int i=10,B=0,temp;
  while(0<i)
  {
temp=(0<i?1:0);
  if(temp!=0)
  {
     i=i-1;
    B=B+2;
    }
    else
    break;
    }
```

c. Assume that the register $t1 is initialized to the value N. How many MIPS instructions are executed?

5N+2