1. Assume that registers $s0 and $s1 hold the values 0x8000_0000 and 0xD000_0000, respectively.

1) What is the value of $t0 for the following assembly code?

Add $t0, $s0, $s1

2) Is the result in $t0 the desired result or has there been overflow?

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

Sub $t0, $s0, $s1

4) Is the result in $t0 the desired result, or has there been overflow?

2. Provide a minimal set of MIPS instructions that may be used to implement the following pseudoinstruction:

Not $t1, $t2

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

A = C[0] << 4;

4. Translate the following C code to MIPS assembly code. Use a minimum number of instructions. Assume that the value of a, b, I, and j are in registers $s0, $s1, $t0, and $t1, respectively. Also, assume that register $s2 holds the base address of the array D.

```
for (i=0; i<a; i++)
    for(j=0; j<b; j++)
        D[4*j] = i+j;
```

5. Implement the following C code in MIPS assembly. What is the total number of MIPS instructions needed to execute the function?

```
int fib(int n)
{
    if (n==0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return fib(n−1) + fib(n−2);
}
```