1. What binary number does this hexadecimal number represent: 7fff fffa_hex? What hexadecimal number does this binary number represent: 1100 1010 1111 1110 1111 1010 1100 1110_two? What decimal number do they represent, respectively?

2. Implement the following C code in MIPS, assuming that set_array is the first function called:

```c
int i;
void set_array(int num) {
    int array[10];
    for (i=0; i<10; i++) {
        array[i] = compare(num, i);
    }
}
int compare(int a, int b) {
    if (sub(a, b) >= 0)
        return 1;
    else
        return 0;
}
int sub (int a, int b) {
    return a-b;
}
```

Be sure to handle the stack and frame pointers appropriately. The variable code font is allocated on the stack, and i corresponds to $s0. Draw the status of the stack before calling set_array and during each function call. Indicate the names of registers and variables stored on the stack and mark the location of $sp and $fp.

3. Add comments to the following MIPS code and describe in one sentence what it computes. Assume that $a0 and $a1 are used for the input and both initially contain the integers a and b, respectively. Assume that $v0 is used for the output.

```mips
add $t0, $zero, $zero
loop: beq $a1, $zero, finish
add $t0, $t0, $a0
addi $a1, $a1, -1
j loop
finish: addi $t0, $t0, 100
add $v0, $t0, $zero
```

4. The following code fragment processes two arrays and produces an important value in
register $v0. Assume that each array consists of 2500 words indexed 0 through 2499, that the base addresses of the arrays are stored in $a0 and $a1 respectively, and their sizes (2500) are stored in $a2 and $a3, respectively. Add comments to the code and describe in one sentence what this code does. Specifically, what will be returned in $v0?

```mips
sll $a2, $a2, 2
sll $a3, $a3, 2
add $v0, $zero, $zero
add $t0, $zero, $zero
outer:
    add $t4, $a0, $t0
    lw $t4, 0($t4)
    add $t1, $zero, $zero
inner:
    add $t3, $a1, $t1
    lw $t3, 0($t3)
    bne $t3, $t4, skip
    addi $v0, $v0, 1
skip:
    addi $t1, $t1, 4
    bne $t1, $a3, inner
    addi $t0, $t0, 4
    bne $t0, $a2, outer
```

5. Show the single MIPS instruction or minimal sequence of instructions for this C statement:

```c
```

Assume that c corresponds to register $t0 and the array x has a base address of 4,000,000
ten.

6. Write a procedure, itoa, in MIPS assembly language that will convert an integer argument into an ASCII decimal string. The procedure should take two arguments: the first is an integer in register $a0; the second is the address at which to write a result string in register $a1. Then itoa should convert its first argument to a null-terminated decimal ASCII string and store that string at the given result location. The return value from itoa, in register $v0, should be a count of the number of non-null characters stored at the destination.