Problem 1: Write the MIPS instructions that produce the following binary code. Interpret the action of each instruction:

   (a) 000000 01001 01000 00000 00000 100000

   (b) 000000 00000 01001 00000 01010 000000

   (c) 000000 00000 00000 00000 00000 000000

Problem 2: Consider a MIPS pseudoinstruction that initializes registers 8 through 15 to 0. How will an assembler expand this into "real" instructions?

Problem 3: The following pseudoinstruction interchanges the contents of registers $r1 and $r2:

                    swap $s1, $s2

How should a MIPS assembler translate it into real instructions?

Problem 4: Implement a pseudoinstruction to copy data from a memory location whose address is in $src register to another memory location whose address is in $dst register:

        mcopy $dst, $src   # M($dst) ← M($src)

Problem 5:

a. Add comments to the following MIPS code to describe its operation. Assume that $a0 and $a1 are used for inputs and initially contain positive integers 6 and 10, respectively. Assume that $v0 is used for output.

```
          add    $t0,   $zero, $zero
loop      beq    $a1,   $zero, exit
          add    $t0,   $t0,   $a0
          addi   $a1,   $a1,   −1
          j      loop
exit      addi   $t0,   $t0,   − 59
          add    $v0,   $t0,   $zero
```

b. What is the final value of the integer in $v0?

Problem 6: A program uses registers 16 through 25. It calls a procedure. Write the instructions that compiler should include in the program before and after the call.