COE 147 Spring 2013
Lab 4 Solution: Simple Functions

Part 1: drawPattern, getPattern, disruptPattern

```
#COE 0147 Lab 4 Part 1 Template

#This template includes testing code, but also has some support code to check
#for a common error.

.text:
        #This is the beginning of the testing code. You should not need to alter
this.

        li $a0, 0xFFFF0008      #LED memory starts at this address
        li $a1, 0x7EF965BD      #Pattern to draw. It will then be disrupted.
        jal drawPattern         #Jump and link to drawPattern, to draw an
                                #initial pattern on the display.

        #Jump and link to disruptPattern. This call should alter the display by
        #disrupting the pattern that was drawn via drawPattern. This will occur
        #so fast that you will not see the original pattern that was drawn.
        li $a0, 0xFFFF0008      #LED memory starts at this address
        jal disruptPattern

        la $a0, successfulQuitMessage
        li $v0, 4
        syscall

        li $v0, 10             #Exit syscall
        syscall

        #This is the end of the testing code.
#========================================
# * Place your drawPattern code here    *
#========================================
    drawPattern:
      sw $a1, ($a0)
      jr $ra


#========================================
# * DO NOT ALTER THIS NEXT LINE         *
j returnErrorHappened
#========================================
#========================================
# * Place your getPattern code here     *
#========================================
    getPattern:
      lw $v0, ($a0)
      jr $ra
```

```
#========================================
# * DO NOT ALTER THIS NEXT LINE         *
j returnErrorHappened
#========================================
#========================================
# * Place your disruptPattern code here *
#========================================
    disruptPattern:
      move $t0,$ra                  #Saves return address into $t7
      jal getPattern
      move $a1,$v0            #Returns return address to $ra
      xori $a1, 0xC31601C9
      jal drawPattern
      move $ra, $t0
      jr $ra


#========================================
# * DO NOT ALTER THIS NEXT LINE         *
j returnErrorHappened
#========================================

returnErrorHappened:
    #If this code is executed, your function did not properly return.
    la $a0, badReturnMessage
    li $v0, 4
    syscall
    li $v0, 10
    syscall

.data:
    badReturnMessage:       .asciiz "A function did not properly return!"
    successfulQuitMessage:  .asciiz "The program has finished."
```

Part 2: drawRepeatedPattern

```
#COE 0147 Lab 4 Part 2 Template

#This template includes testing code, but also has some support code to check
#for a common error.

.text:
        #This is the beginning of the testing code.

      # you may put additional instructions to calculate the addresses and bit
patterns

      li $a0, 0xFFFF0008      # replace your_address with the actual address
      li $a1, 0x7EF965BD      # replace your_pattern with the actual pattern
      li $a2, 5               #Draw the pattern 5 times Vertically.
      jal drawRepeatedPattern #Jump and link to drawRepeatedPattern.


      li $a0, 0xFFFF0036      # replace your_address with the actual address
      li $a1, 0xBDEF7464      # replace your_pattern with the actual pattern
      li $a2, 7               #Draw the pattern 7 times vertically.
      jal drawDiagonalPattern #Jump and link to drawRepeatedPattern.

      # do not alter
      la $a0, successfulQuitMessage
      li $v0, 4
      syscall

      li $v0, 10             #Exit syscall
      syscall

      #This is the end of the testing code.

#========================================
# * Place your drawPattern code here    *
#========================================
    drawPattern:
      sw $a1, ($a0)
      jr $ra



#========================================
# * DO NOT ALTER THIS NEXT LINE         *
j returnErrorHappened
#========================================
#========================================
# * Place drawRepeatedPattern code here *
#========================================
    drawRepeatedPattern:
      move $t0,$ra

    LOOP:
      beq $a2,$zero,EXIT
      jal drawPattern
      la $a0, 32($a0)
      addi $a2,$a2,-1
      j LOOP

    EXIT:
      move $ra, $t0
      jr $ra

#========================================
# * DO NOT ALTER THIS NEXT LINE         *
j returnErrorHappened
#========================================

returnErrorHappened:
    #If this code is executed, your function did not properly return.
    la $a0, badReturnMessage
    li $v0, 4
    syscall
    li $v0, 10
    syscall
```

```
.data:
    badReturnMessage:       .asciiz "A function did not properly return!"
    successfulQuitMessage:  .asciiz "The program has finished."
```