COE 0147 Spring 2013
Lab 2 Solution: Immediate Values, Memory, and System Calls

Part 1: Immediate Values

```
.data:

#Question 1: What is the machine code (in hexadecimal) of these instructions?
#Sequence 1
# - Machine Code 1: 0x3C09DEAD
# - Machine Code 2: 0x3529ACFB
#Sequence 2
# - Machine Code 1: 0x3409DEAD
# - Machine Code 2: 0x00094C00
# - Machine Code 3: 0x3529ACFB

#Question 2: What instruction format are these instructions (R, I, or J)?
#Sequence 1: I
#Sequence 2: I

#Question 3: What are the values (in hexadecimal) of the immediate field in each
instructions?
#Sequence 1
# - Immediate Field 1: DEAD
# - Immediate Field 2: ACFB
#Sequence 2
# - Immediate Field 1: DEAD
# - Immediate Field 2: n/a
# - Immediate Field 3: ACFB

.text:
#=============================================================
#Place here your first instructions to put 0xDEADACFB into $t1.

lui $t1, 0xDEAD
ori $t1, $t1, 0xACFB


#=============================================================

#=============================================================
#Place here your second instructions to put 0xDEADACFB into $t1
#There were a number of solutions for doing this, this is one example

li   $t1, 0xDEAD
sll  $t1, 16
ori  $t1, $t1, 0xACFB

#=============================================================

#=============================================================
#The rest of the template code goes here.
```

Part 2: Memory

```
.data

x: .half 7
y: .half 17
z: .half 0

.text

la $t0, x #Address of x is now in $t0
lh $s0, 0($t0) #Value of x is in $s0
lh $s1, 2($t0) #Value of y is in $s1

#compute z #z = x + y
add $s2, $s0, $s1

sh $s2, 4($t0) #Store z's new value to z's address
               #This will overwrite the old value
sh $s2, 2($t0) #Store z's new value to y's address
sh $s2, 0($t0) #Store z's new value to x's address
```

Part 3: System Calls

```
.data

prompt1:        .asciiz "What is the first value?\n"
prompt2:        .asciiz "What is the second value?\n"
result_part_1:  .asciiz "The difference of "
and_str:        .asciiz " and "
is_str:         .asciiz " is "

.text

#print "What is the first value?"
la $a0, prompt1
li $v0, 4
syscall

li $v0, 5 #Read integer syscall
syscall

#Now, the first number is in $v0.
#Copy it $v0's value to $s0 for safe keeping.
move $s0, $v0

#print "What is the second value?"
la $a0, prompt2
li $v0, 4
syscall

li $v0, 5 #Read integer syscall
syscall

#Now, the second number is in $v0.
#Copy it $v0's value to $s1 for safe keeping.
move $s1, $v0

#print "The difference of..."
la $a0, result_part_1
li $v0, 4
syscall

#Print out the first number.
move $a0, $s0
li $v0, 1
syscall

#print " and "
la $a0, and_str
li $v0, 4
syscall

#print out the first number.
move $a0, $s1
li $v0, 1
syscall

#print " is "
la $a0, is_str
li $v0, 4
syscall
```

```
#Subtract $s1 from $s0 and put the result in $a0.
sub $a0, $s0, $s1
li $v0, 1
syscall
```