Ramzi El-abdallah
February 24, 2014
CS 231

Assignment 1

1) -26 – 35= 26: 0000 0000 0001 1010
1111 1111 1110 0101 (1's compliment)
1 (add 1)
-26: 1111 1111 1110 0110

35: 0000 0000 0010 0011
1111 1111 1101 1100 (1's compliment)
1 (add 1)

-35: 1111 1111 1101 1101
-26: 1111 1111 1110 0110
1111 1111 1100 0011 (result = -61)

2) What the following code is doing is giving us the square of the input divided by 2. If the input is odd then if it divides the input by 2 and rounds up and gives us the square of that number.

For Example:

INPUT	OUTPUT
1,2	1
3,4	4
5,6	9
7,8	16

3)

```
# Assignment 1 - Question 3

        .data
location1: .word 45, 67        #first memory location
location2: .word 78, 90        #second memory location

        .text

main:
        la $s0, location1      #loads the address of location1 into $s0
        la $s1, location2      #loads the address of location2 into $s1

        lw $t0, 0($s0)         #loads the first number of location1 (45) into $t0
        lw $t1, 0($s1)         #loads the first number of location2 (78) into $t1
```

```
        sw $t0, 0($s1)      #replaces first number of location2 with first number of location1
        sw $t1, 0($s0)      #replaces first number of location1 with first number of location2

        lw $t0, 4($s0)      #loads the second number of location1 (67) into $t0
        lw $t1, 4($s1)      #loads the second number of location2 (90) into $t1

        sw $t0, 4($s1)      #replaces second number of location2 with second number of location1
        sw $t1, 4($s0)      #replaces second number of location1 with second number of location2

        li $v0, 10          #exit
        syscall

# END OF PROGRAM
```

4) Convert to hexadecimal:

0000 0001 1111 1000 0010 0001 1100 1111 =
0-1-15-8-2-1-12-15 (decimal representation for every 4 bits)

Hexadecimal Representation: 01E821CE

5)
```
li $s1, 4000000
lw $s0, 44($s1)
add $t0, $s0, $t0
sw $t0, 40($s1)
```