Name: KEY **Id#**

COE 301/ICS 233, Term 151

Computer Architecture & Assembly Language

Quiz# 3

Date: Sunday, Oct. 25, 2015

Q1. Write a procedure, **GCD**, that receives two positive numbers in $a0 and $a1 and returns their greatest common divisor in register $v0. It is required that the procedure **preserves the content of all used registers** according to the MIPS programming convention by saving them and restoring them on the stack. The pseudo code of the GCD procedure is given below:

```
int gcd(int m, int n) {
   if ((m % n) == 0)
      return n;
   else
      return gcd(n, m % n);
}
```

```
GCD:
    divu $a0, $a1
    mfhi $t0
    bne $t0, $0, Else
    move $v0, $a1
    jr $ra
    Else:
    addi $sp, $sp, -4
    sw $ra, ($sp)
    move $a0, $a1
    move $a1, $t0
    jal GCD
    lw $ra, ($sp)
    addi $sp, $sp, 4
    jr $ra
```

Q2.

(i) Given that **Multiplicand=1010** and **Multiplier=1011,** using **signed multiplication**, show the **signed** multiplication of **Multiplicand** by **Multiplier**. The result of the multiplication should be an 8 bit **signed** number in HI and LO registers. Show the steps of your work.

Iteration		**Multiplicand**	**Sign**	**Product = HI,LO**
0	Initialize (LO = Multiplier)	1010		0000 1011
1	LO[0] = 1 => ADD		1	1010 1011
	Shift Product = (HI, LO) right 1 bit	1010		1101 0101
2	LO[0] = 1 => ADD		1	0111 0101
	Shift Product = (HI, LO) right 1 bit	1010		1011 1010
3	LO[0] = 0 => Do nothing		1	1011 1010
	Shift Product = (HI, LO) right 1 bit	1010		1101 1101
4	LO[0] = 1 => SUB		0	0011 1101
	Shift Product = (HI, LO) right 1 bit			**0001 1110**

(ii) Given that **Dividend=1011** and **Divisor=0010**, Using **unsigned division**, show the **unsigned** division of **Dividend** by **Divisor**. The result of division should be stored in the Remainder and Quotient registers. Show the steps of your work.

Iteration		**Remainder (HI)**	**Quotient (LO)**	**Divisor**	**Difference**
0	Initialize	0000	1011	0010	
1	1: SLL, Difference	0001	0110	0010	1111
	2: Diff < 0 => Do Nothing	0001	0110	0010	
2	1: SLL, Difference	0010	1100	0010	0000
	2: Rem = Diff, set lsb Quotient	0000	1101	0010	
3	1: SLL, Difference	0001	1010	0010	1111
	2: Diff < 0 => Do Nothing	0001	1010	0010	
4	1: SLL, Difference	0011	0100	0010	0001
	2: Rem = Diff, set lsb Quotient	**0001**	**0101**	0010	