**Test Cases**

**1 - Normal Test Cases :**

**1) Assembly Format Binary Format**

add $s0,$s1,$s2 00000010001100101000000000100000

sub $s3,$s4,$s5 00000010100101011001100000100010

ori $s1,$s1,10 00110110001100010000000000001010

addi $s2,$s2,10 00100010010100100000000000001010

sll $s4,$s4,2 00000000000101001010000010000000

sra $s5,$s5,2 00000000000101011010100010000011

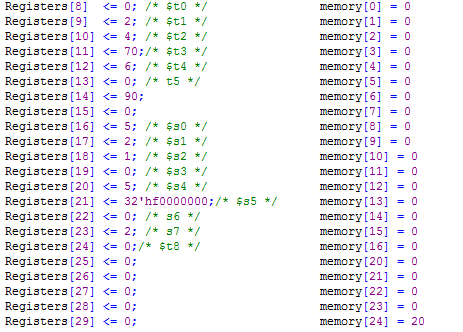
slt $s6,$s7,$s0 00000010111100001011000000101010

andi $t0,$t1,10 00110001001010000000000000001010

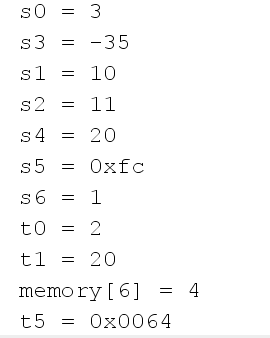
lw $t1,20($t2) 10001101010010010000000000010100

sw $t2,0($t4) 10101101100010100000000000000000

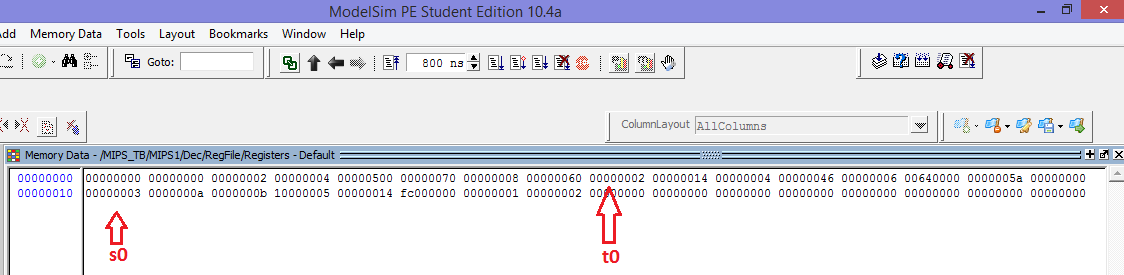
lui $t5,100 00111100000011010000000001100100



**Expected outputs :**

****

**Actual outputs :**

****

Memory :

C:\Users\abdulrahman\Desktop\Capture1.PNG

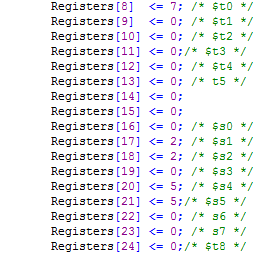
**Testing Branch and Jump :**

**2)Assembly Format**  **Binary Format**

InstrucionMem[2] =beq $s1,$s2,label 00010010001100100000000000001010

InstrucionMem[3]= add $s3,$s4,$t0 00000010100010001001100000100000

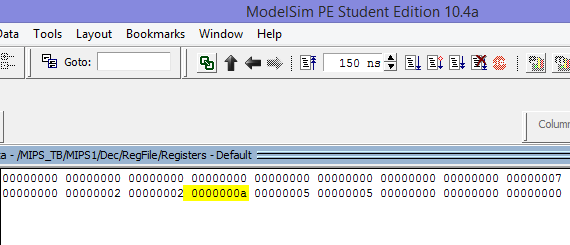
InstrucionMem[14]= label : add $s3,$s4,$s5 00000010100101011001100000100000



**Expected outputs :**

s3 = 10

**Actual outputs :**

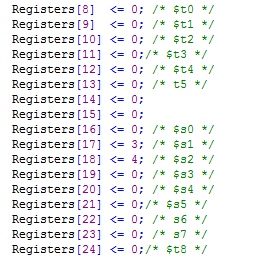


**3) Assembly Format**  **Binary Format**

InstrucionMem[0] = bnq $s1,$s2,label 00010110001100100000000000001101

InstrucionMem[1] = add $s3,$s1,$s2 00000010001100101001100000100000

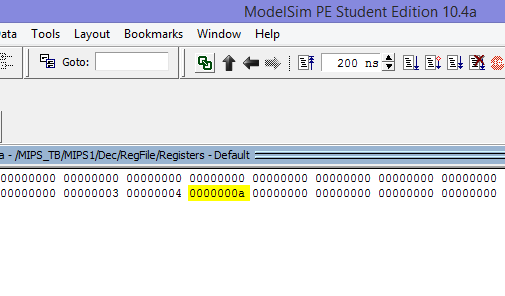
InstrucionMem[15] = label : addi $s3,$s3,10 00100010011100110000000000001010



**Expected outputs :**

s3 = 10

**Actual outputs :**

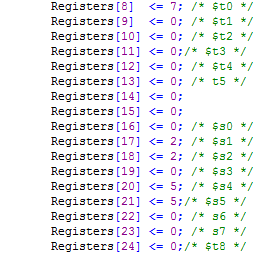
****

**4)Assembly Format**  **Binary Format**

InstrucionMem[1]= j label 00001000000000000000000000010011

InstrucionMem[2]=add $s3,$s4,$t0 00000010100010001001100000100000

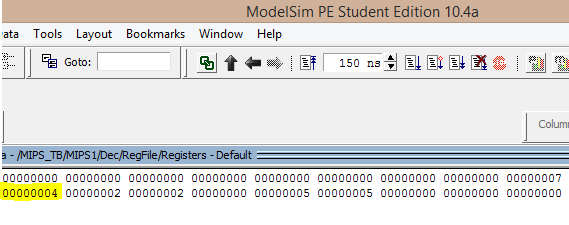
InstrucionMem[21] = label : add $s0,$s1,$s2 00000010001100101000000000100000



**Expected outputs :**

s0 = 4, s3 = 0

**Actual outputs :**

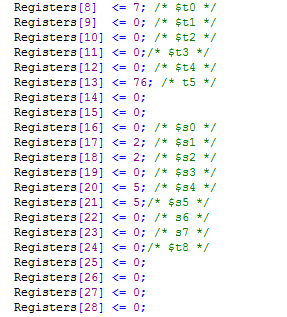


**5) Assembly Format**  **Binary Format**

InstrucionMem[1] = jr $t5 00000001101000000000000000001000

InstrucionMem[2]= add $t2,$s1,$s2 00000010001100100101000000100000

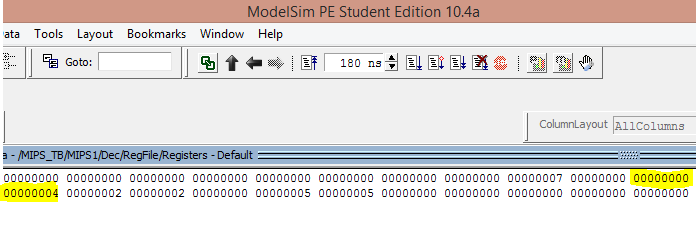
InstrucionMem[19] = add $s0,$s1,$s2 00000010001100101000000000100000



**Expected outputs :**

s0 = 4, t2 = 0

**Actual outputs :**

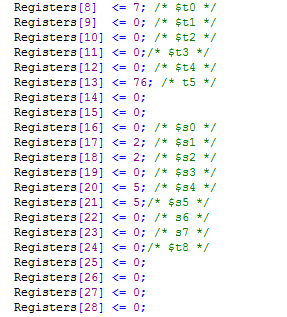
****

**6) Assembly Format**  **Binary Format**

InstrucionMem[1]= jal L 00001100000000000000000000010011

InstrucionMem[2]= add $t2,$s1,$s2 00000010001100100101000000100000

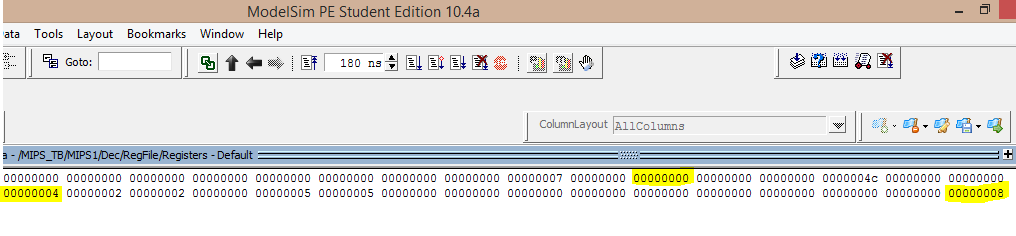
InstrucionMem[21]= L: add $s0,$s1,$s2 00000010001100101000000000100000

****

**Expected outputs :**

s0 = 4, t2 = 0, ra = 8

**Actual outputs :**

****

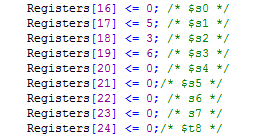
**2 - Test Cases with Data Hazards :**

**1) Assembly Format**  **Binary Format**

InstrucionMem[1] = add $s0,$s2,$s1 00000010010100011000000000100000

InstrucionMem[2] =add $s4,$s0,$s3 00000010001100100101000000100000

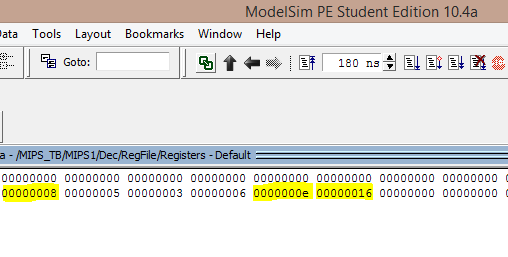
InstrucionMem[3] = add $s5,$s4,$s0 00000010001100101000000000100000



**Expected outputs :**

s0 = 8, s4 = 14, s5 = 24

**Actual outputs :**

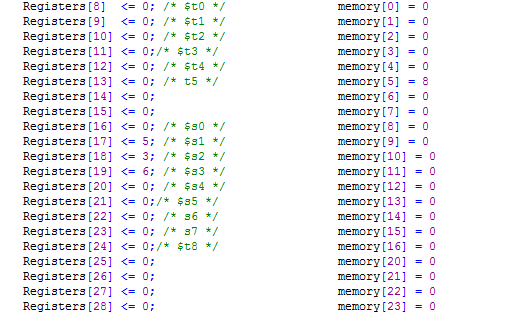
****

**2) Assembly Format**  **Binary Format**

InstrucionMem[1] = lw $s0,0($s1) 10001110001100000000000000000000

InstrucionMem[2] = add $s4,$s0,$s3 00000010001100100101000000100000

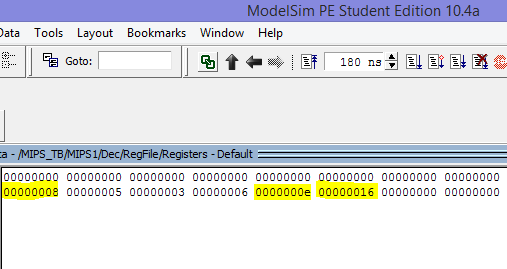
InstrucionMem[3] = add $s5,$s4,$s0 00000010001100101000000000100000



**Expected outputs :**

s0 = 8, s4 = 14, s5 = 24

**Actual outputs :**

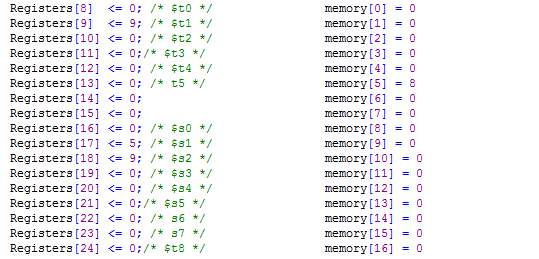


**3) Assembly Format**  **Binary Format**

InstrucionMem[1] = lw $s0,0($s1) 10001110001100000000000000000000

InstrucionMem[2] = sw $s2,0($s0) 10101110000100100000000000000000

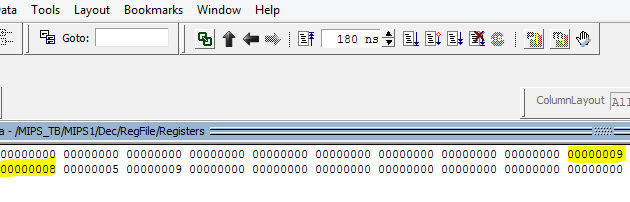
InstrucionMem[3] = lw $t0,0($t1) 10001101001010000000000000000000



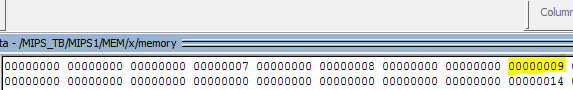
**Expected outputs :**

s0 = 8, t0 = 9, memory[8]=9

**Actual outputs :**

****

Memory:

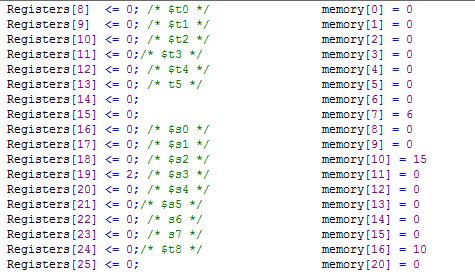


**4) Assembly Format**  **Binary Format**

InstrucionMem[1] = lw $s1, 5($s3) 10001110011100010000000000000101

InstrucionMem[2] = lw $t0, 10($s1) 10001110001010000000000000001010

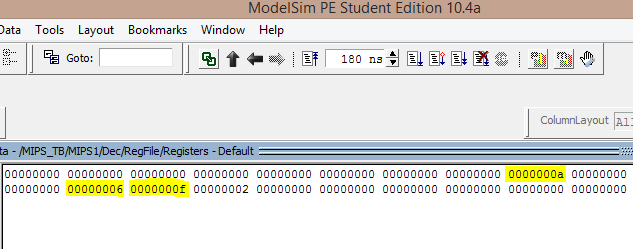
InstrucionMem[3] = lw $s2, 0($t0) 10001101000100100000000000000000



**Expected outputs :**

s1 = 6, t0 = 10, s2=15

**Actual outputs :**

****

**3 - Test Cases with Control Hazards :**

**1) Assembly Format**  **Binary Format**

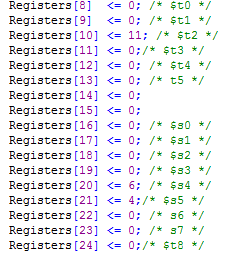
InstrucionMem[0] = add $s2,$s4,$s5 00000010100101011001000000100000

InstrucionMem[1] = add $s1,$s2,$s6 00000010010101101000100000100000

InstrucionMem[2] = beq $s1,$s2,Label 00010010001100100000000000001010

InstrucionMem[3] = add $t0,$s1,$t2 00000001001010100100000000100000

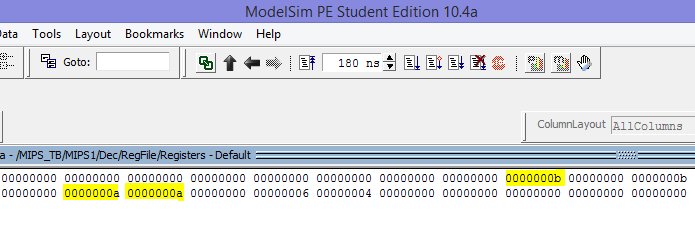
Label :InstrucionMem[13] = add $t0,$t0,$t2 00000001000010100100000000100000



**Expected outputs :**

s1 = 10, s2 = 10, t0=11

**Actual outputs :**

****

**2) Assembly Format**  **Binary Format**

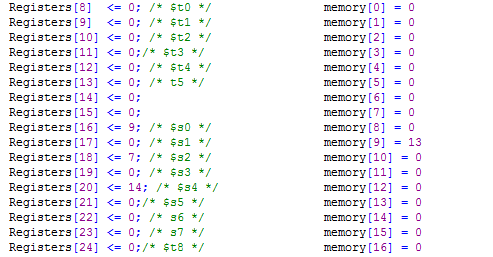
InstrucionMem[0] = lw $t1, 0($s0) 10001110000010010000000000000000

InstrucionMem[1] = add $t2,$t1,$s3 00000001001100110101000000100000

InstrucionMem[2] = beq $t1,$t2,label 00010001001010100000000000010000

InstrucionMem[3] = add $s1,$s3,$s4 00000010011101001000100000100000

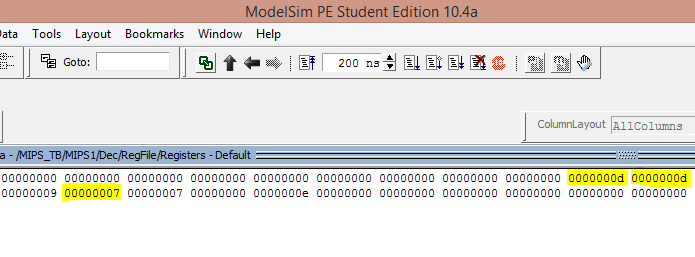
Label :InstrucionMem[19] = add $s1,$s2,$s3 00000010010100111000100000100000



**Expected outputs :**

t1 = 13, t2 = 13, s1=7

**Actual outputs :**

****

**3) Assembly Format**  **Binary Format**

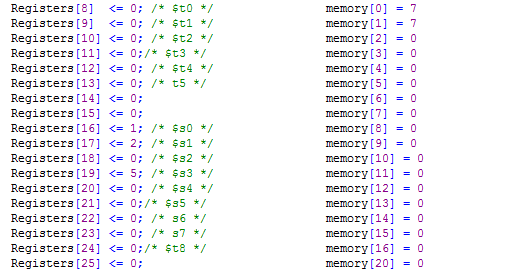
InstrucionMem[0] = lw $t1, 0($s0) 10001110000010010000000000000000

InstrucionMem[1] = lw $t2, 0($s1) 10001110001010100000000000000000

InstrucionMem[2] = beq $t1,$t2,label 00010001001010100000000000001010

InstrucionMem[3] = add $s2,$s3,$s3 00000010011100111000100000100000

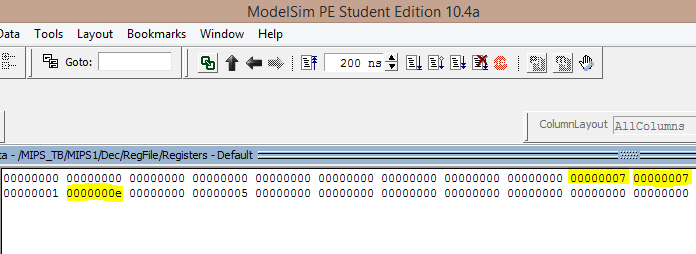
Label :InstrucionMem[13] = add $s1,$t1,$t2 00000001001010101000100000100000



**Expected outputs :**

t1 = 7, t2 = 7, s1=14

**Actual outputs :**

****

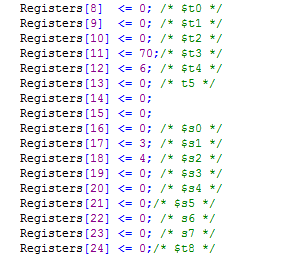
**4) Assembly Format**  **Binary Format**

InstrucionMem[1] = add $t5,$t3,$t4 00000001011011000110100000100000

InstrucionMem[2] = jr $t5 00000001101000000000000000001000

InstrucionMem[3]= add $t2,$s1,$s2 00000010001100100101000000100000

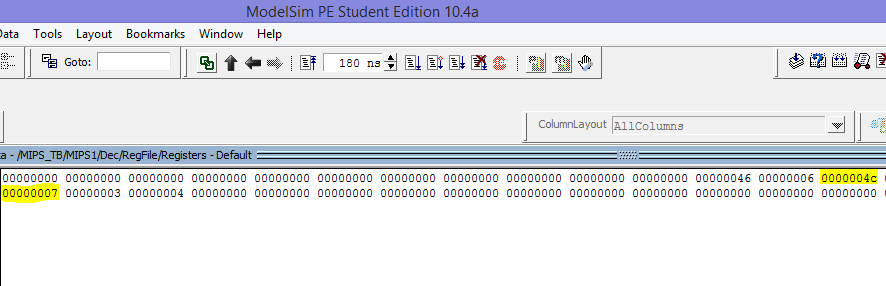
InstrucionMem[20] = add $s0,$s1,$s2 00000010001100101000000000100000



**Expected outputs :**

t5 = 76, s0=7

**Actual outputs :**

****

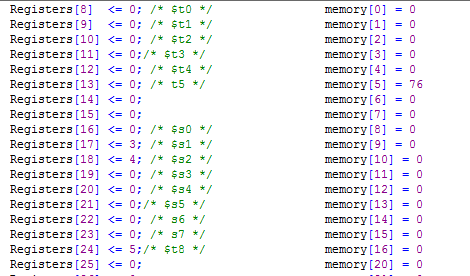
**5) Assembly Format**  **Binary Format**

InstrucionMem[1] = lw $t5,0($t8) 10001111000011010000000000000000

InstrucionMem[2] = jr $t5 00000001101000000000000000001000

InstrucionMem[3] = add $t2,$s1,$s2 00000010001100100101000000100000

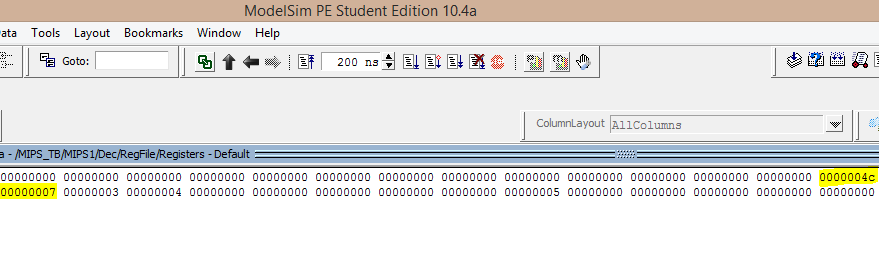
InstrucionMem[20] = add $s0,$s1,$s2 00000010001100101000000000100000



**Expected outputs :**

t5 = 76, s0=7

**Actual outputs :**

****

**1 –General Test with (or,ori,xor) :**

**1) Assembly Format**  **Binary Format**

xor $s0,$s1,$s2 00000010001100101000000000100110

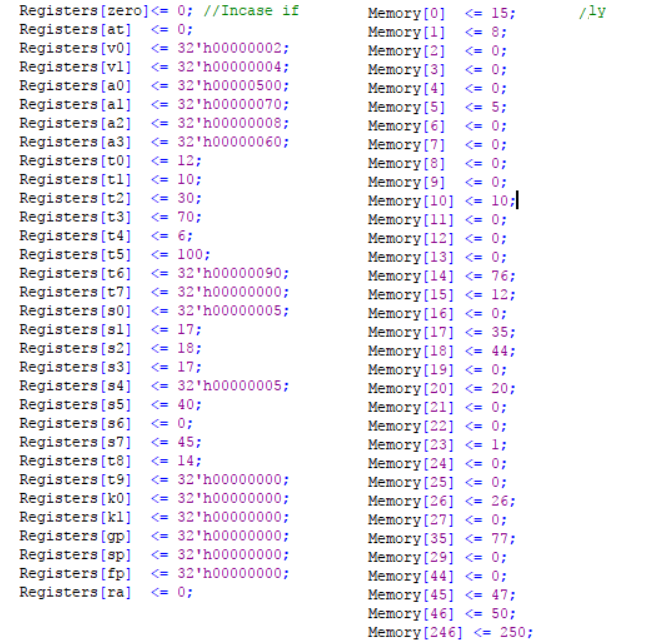
ori $t1,$t2,15 00110101010010010000000000001111

or $t3,$t4,$t5 00000001100011010101100000100101

lw $s3,4($t8) 10001111000100110000000000000100

and $s6,$s3,$s4 00000010011101001011000000100100

add $s6,$zero,$s6 00000000000101101011000000100000



**Expected outputs :**

**S0=3**

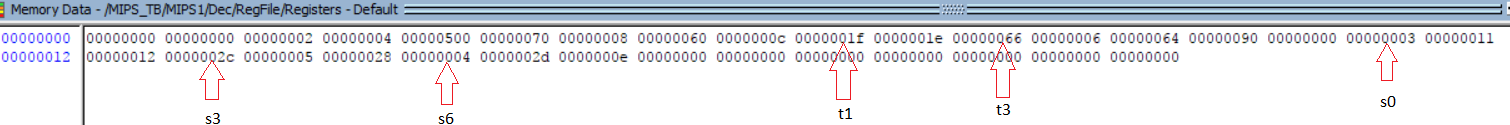
**T1=31**

**T3=102**

**S3=44**

**S6=4**

**Actual outputs :**

****