ECE 346 HW # 1

Bradley McKee

UIN: 00975338

Question 1)

lui $s0, 0x1001

op	rs	rt	imm
001111	00000	10000	0001000000000001

xori $s1, $s0, -10

op	rs	rt	imm
001110	10000	10001	1111111111110111

lui $s2,0x222

op	rs	rt	imm
001111	00000	10010	0000001000100010

add $s3,$s1,$s2

op	rs	rt	rd	sham	funct
000000	10001	10010	10011	00000	100000

sw $s3,4($s0)

op	rs	rt	im
101011	10000	10011	0000000000000100

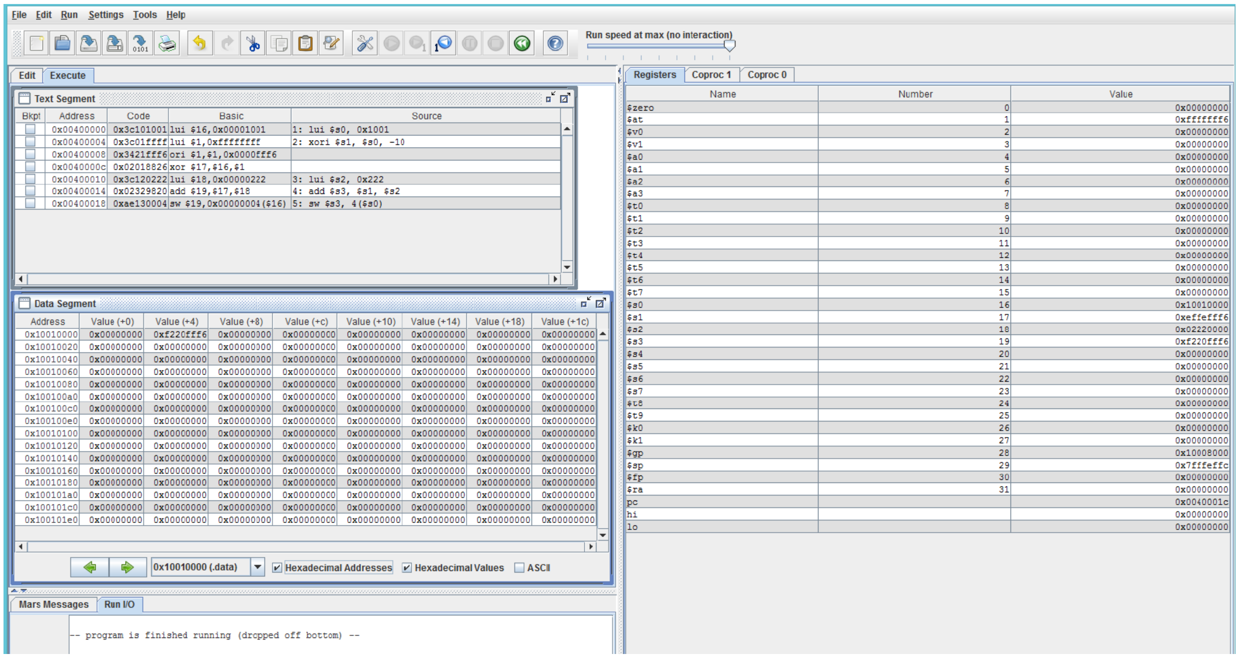
1.2)

Function Type	Register	address
lui	$s0	0x10010000
xori	$s1	0x1000FFF7
lui	$s2	0x02220000
add	$s3	0x1222FFF7
sw	all	take contents of $s3 offset it by 4 bits of the base address ($s0)

Final addresses:

$s0 0x00010000

$s1	0x1000FFF7
$s2	0x02220000
$s3	0x1222FFF7



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x3c101001	lui $16,0x00001001	1: lui $s0, 0x1001
□	0x00400004	0x3c01ffff	lui $1,0xffffffff	2: xori $s1, $s0, -10
□	0x00400008	0x3421fff6	ori $1,$1,0x0000fff6	
□	0x0040000c	0x02018826	xor $17,$16,$1	
□	0x00400010	0x3c120222	lui $18,0x00000222	3: lui $s2, 0x222
□	0x00400014	0x02329820	add $19,$17,$18	4: add $s3, $s1, $s2
□	0x00400018	0xae130004	sw $19,0x00000004($16)	5: sw $s3, 4($s0)

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0xf220fff6	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0x00000000
$at	1	0xfffffff6
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x10010000
$s1	17	0xeffefff6
$s2	18	0x02220000
$s3	19	0xf220fff6
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040001c
hi		0x00000000
lo		0x00000000

```
-- program is finished running (dropped off bottom) --
```

1.3) The code doesn’t match from my handwork, from what I noticed it had to break up the xori into 2 commands

1.4) after the program was done, $at is equivalent to $s3 which makes me believe that the register $at was kind of used as a temporary when executing through the different instructions.

Question 2)

0x20080000

0010 00|00 000|0 1000| 0000 0000 0000 0000

op rs rt imm

addi $t0,0

0x20090001

0010 00|00 000|0 1001| 0000 0000 0000 0001

op rs rt imm

addi $t1,0x1

0x0089502A

0000 00|00 100|0 1001| 0101 0|000 00|01 1010 op rs rt rd sham func

SLT $t2,$a0,$t1

0X15400003

0001 01|01 010|0 0000| 0000 0000 0000 0011

op rs rt imm

BNE $t2, $0,0x3

0X01094020

0000 00|01 000|0 1001| 0100 0|000 00|10 0000

Op rs rt rd sham func

add $t0, $t0, $t1

0X21290002

0010 00|01 001|0 1001| 0000 0000 0000 0010

Op rs rt imm

addi $t1,$t1,0x2

0X08100002

0000 10|00 0001 0000 0000 0000 0000 00010

op jump target address

j 0x00400008

0X01001020

0000 00|01 000|0 0000| 0001 0|000 00|10 0000

op rs rt rd shamt func

add $v0, $t0,$0

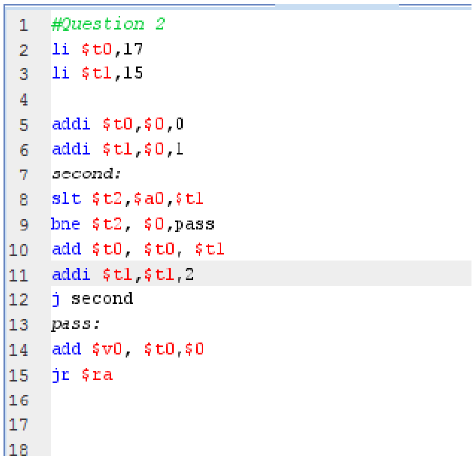
0X03E00008

0000 00|11 111|0 0000| 0000 0|000 00|00 1000

op rs r t rd shamt func

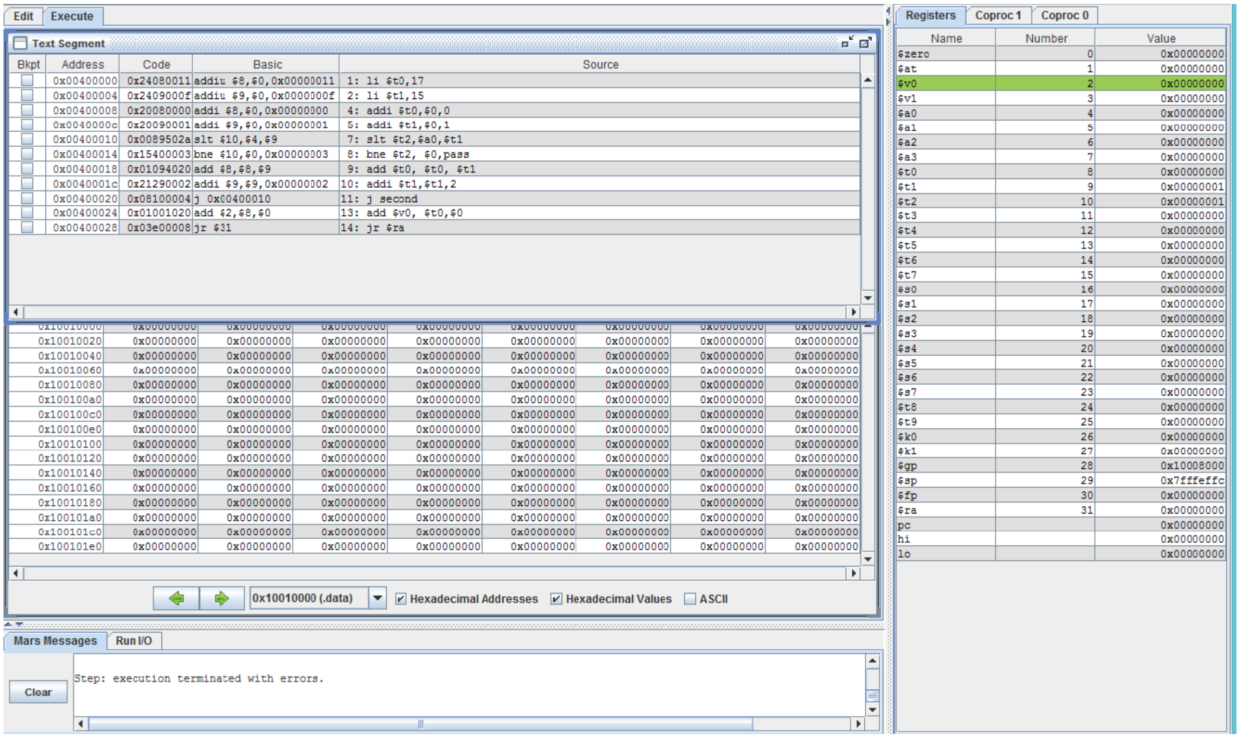
jr $ra

2.3)



```
#Question 2
li $t0,17
li $t1,15

addi $t0,$0,0
addi $t1,$0,1
second:
slt $t2,$a0,$t1
bne $t2, $0,pass
add $t0, $t0, $t1
addi $t1,$t1,2
j second
pass:
add $v0, $t0,$0
jr $ra
```



Bkpt	Address	Code	Basic	Source
□	0x00400000	0x24080011	addiu $8,$0,0x00000011	1: li $t0,17
□	0x00400004	0x2409000f	addiu $9,$0,0x0000000f	2: li $t1,15
□	0x00400008	0x20080000	addi $8,$0,0x00000000	4: addi $t0,$0,0
□	0x0040000c	0x20090001	addi $9,$0,0x00000001	5: addi $t1,$0,1
□	0x00400010	0x0089502a	slt $10,$4,$9	7: slt $t2,$a0,$t1
□	0x00400014	0x15400003	bne $10,$0,0x00000003	8: bne $t2, $0,pass
□	0x00400018	0x01094020	add $8,$8,$9	9: add $t0, $t0, $t1
□	0x0040001c	0x21290002	addi $9,$9,0x00000002	10: addi $t1,$t1,2
□	0x00400020	0x08100004	j 0x00400010	11: j second
□	0x00400024	0x01001020	add $2,$8,$0	13: add $v0, $t0,$0
□	0x00400028	0x03e00008	jr $31	14: jr $ra

0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values □ ASCII

```
Step: execution terminated with errors.
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000001
$t2	10	0x00000001
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00000000
hi		0x00000000
lo		0x00000000

The functionality of this program is to iterate through the values and when register $t2 is not equal to the value in register $0 (0), then it will pass to the adding line. Lastly, it will then jump to the register $ra then the program will terminate.

3a)

```
slt $t0,$s1,$s0
beq $t0,$0, ELSE
add $t1,$s0,$s1
J ENDPROGRAM
ELSE:
Sub $t1,$s0,$s1
ENDPROGRAM:
```

3b)

```
slt $t0,$s1,$s0
beq $s1,$s0, GLABEL
beq $t0,$0, HLABEL
GLABEL:
addi $s0,$s0,1
J ENDPROGRAM
HLABEL:
subi $s1,$s1,1
ENDPROGRAM:
```

3c)

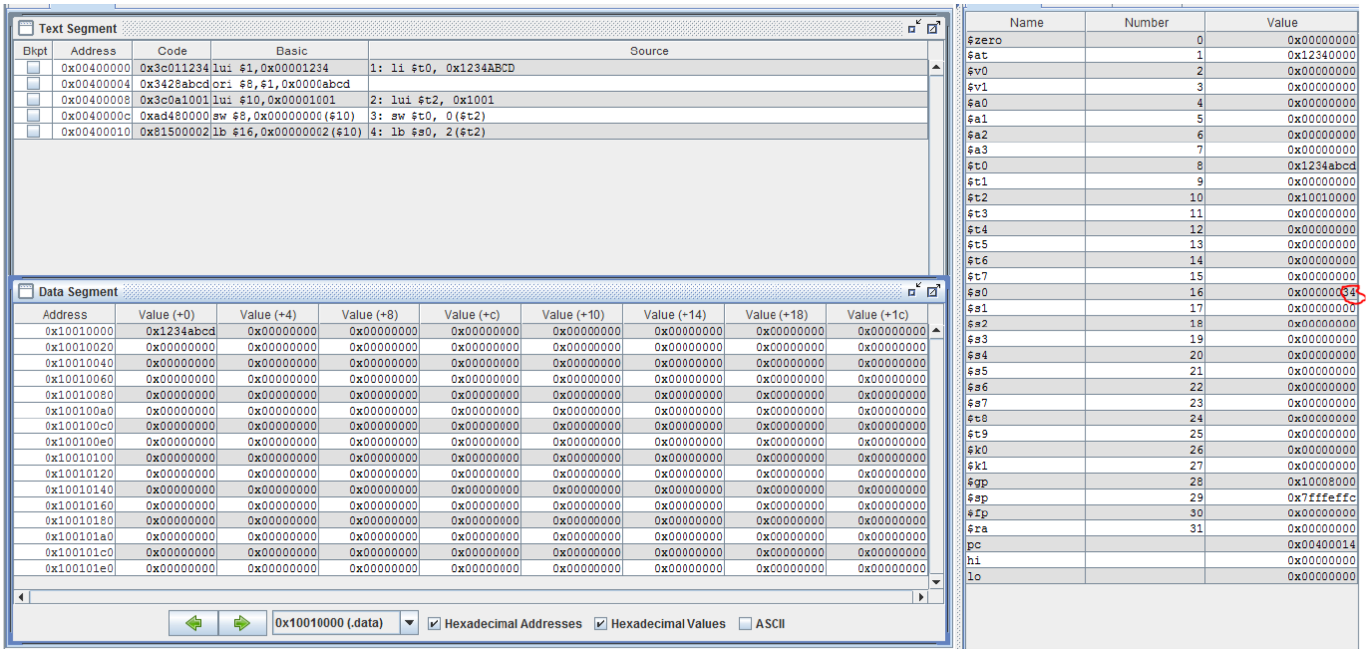
```
slt $t0, $s0,$s1
beq $s0,$s1, GLABEL
beq $t0,$0, HLABEL
GLABEL:
Add $s0, $0, $0
J ENDPROGRAM
HLABEL:
add $s1,$0, $0
ENDPROGRAM:
```

Question 4)

This will produce the output of the second byte from the stored address in $t2.   Big-endian would be AB and little-endian would be 34.

4.2)

From the MIPS Simulation, it says that the second byte is 34, which would make MIPS a little-endian architecture.



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x3c011234	lui $1,0x00001234	1: li $t0, 0x1234ABCD
☐	0x00400004	0x3428abcd	ori $8,$1,0x0000abcd	
☐	0x00400008	0x3c0a1001	lui $10,0x00001001	2: lui $t2, 0x1001
☐	0x0040000c	0xad480000	sw $8,0x00000000($10)	3: sw $t0, 0($t2)
☐	0x00400010	0x81500002	lb $16,0x00000002($10)	4: lb $s0, 2($t2)

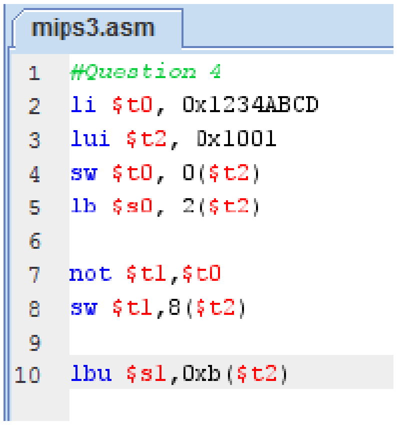
Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x1234abcd	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Name	Number	Value
$zero	0	0x00000000
$at	1	0x12340000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x1234abcd
$t1	9	0x00000000
$t2	10	0x10010000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000034
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400014
hi		0x00000000
lo		0x00000000

4.3a/b)

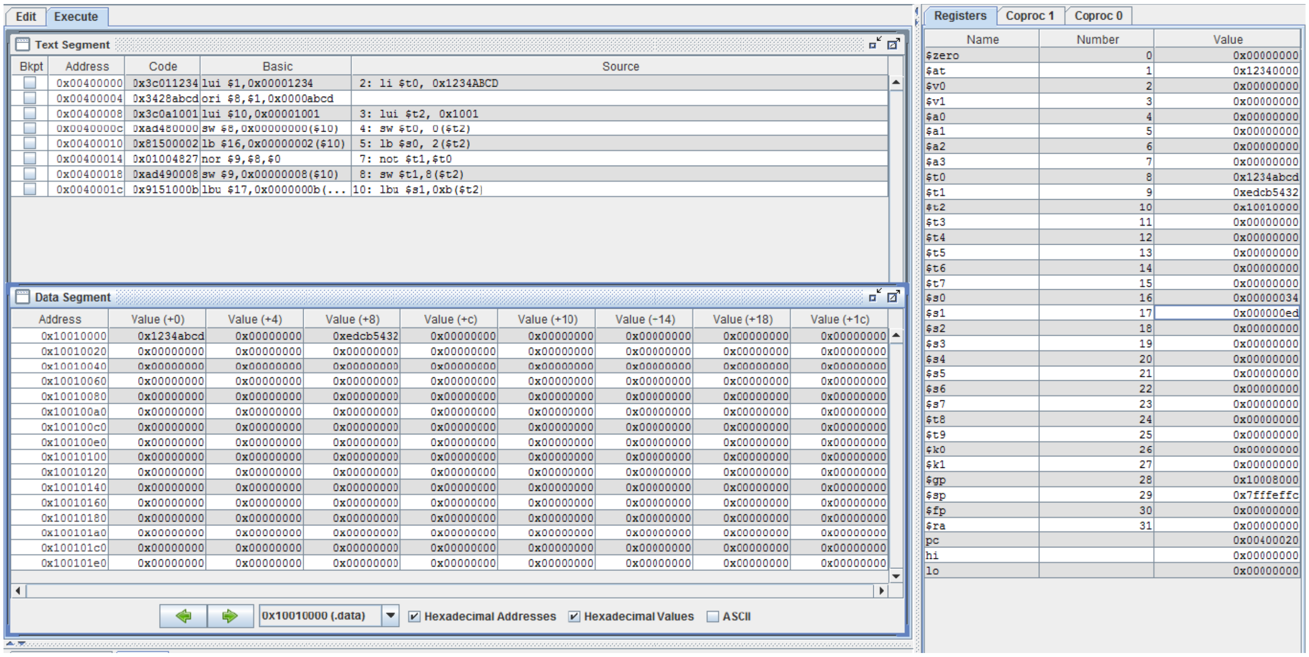


mips3.asm

```
#Question 4
li $t0, 0x1234ABCD
lui $t2, 0x1001
sw $t0, 0($t2)
lb $s0, 2($t2)

not $t1,$t0
sw $t1,8($t2)

lbu $s1,0xb($t2)
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x3c011234	lui $1,0x00001234	2: li $t0, 0x1234ABCD
☐	0x00400004	0x3428abcd	ori $8,$1,0x0000abcd	
☐	0x00400008	0x3c0a1001	lui $10,0x00001001	3: lui $t2, 0x1001
☐	0x0040000c	0xad480000	sw $8,0x00000000($10)	4: sw $t0, 0($t2)
☐	0x00400010	0x81500002	lb $16,0x00000002($10)	5: lb $s0, 2($t2)
☐	0x00400014	0x01004827	nor $9,$8,$0	7: not $t1,$t0
☐	0x00400018	0xad490008	sw $9,0x00000008($10)	8: sw $t1,8($t2)
☐	0x0040001c	0x9151000b	lbu $17,0x0000000b(...	10: lbu $s1,0xb($t2)

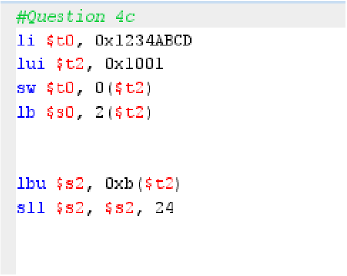
Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (-14)	Value (+18)	Value (+1c)
0x10010000	0x1234abcd	0x00000000	0xedcb5432	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

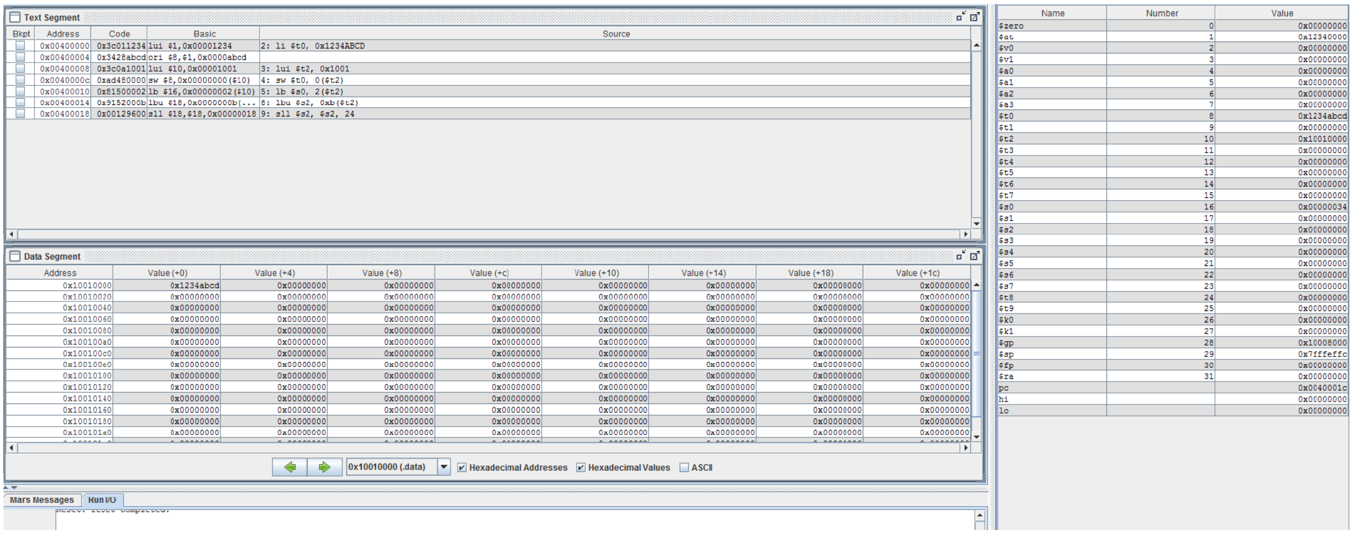
Name	Number	Value
$zero	0	0x00000000
$at	1	0x12340000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x1234abcd
$t1	9	0xedcb5432
$t2	10	0x10010000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000034
$s1	17	0x000000ed
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400020
hi		0x00000000
lo		0x00000000

4.3c)



```
#Question 4c
li $t0, 0x1234ABCD
lui $t2, 0x1001
sw $t0, 0($t2)
lb $s0, 2($t2)


lbu $s2, 0xb($t2)
sll $s2, $s2, 24
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x3c011234	lui $1,0x00001234	2: li $t0, 0x1234ABCD
☐	0x00400004	0x3428abcd	ori $8,$1,0x0000abcd	
☐	0x00400008	0x3c0a1001	lui $10,0x00001001	3: lui $t2, 0x1001
☐	0x0040000c	0xad480000	sw $8,0x00000000($10)	4: sw $t0, 0($t2)
☐	0x00400010	0x81500002	lb $16,0x00000002($10)	5: lb $s0, 2($t2)
☐	0x00400014	0x9152000b	lbu $18,0x0000000b(...	8: lbu $s2, 0xb($t2)
☐	0x00400018	0x00129600	sll $18,$18,0x00000018	9: sll $s2, $s2, 24

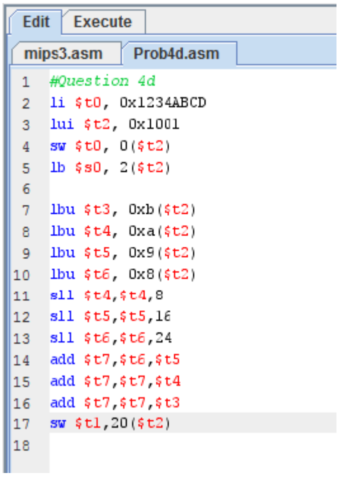
Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x1234abcd	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

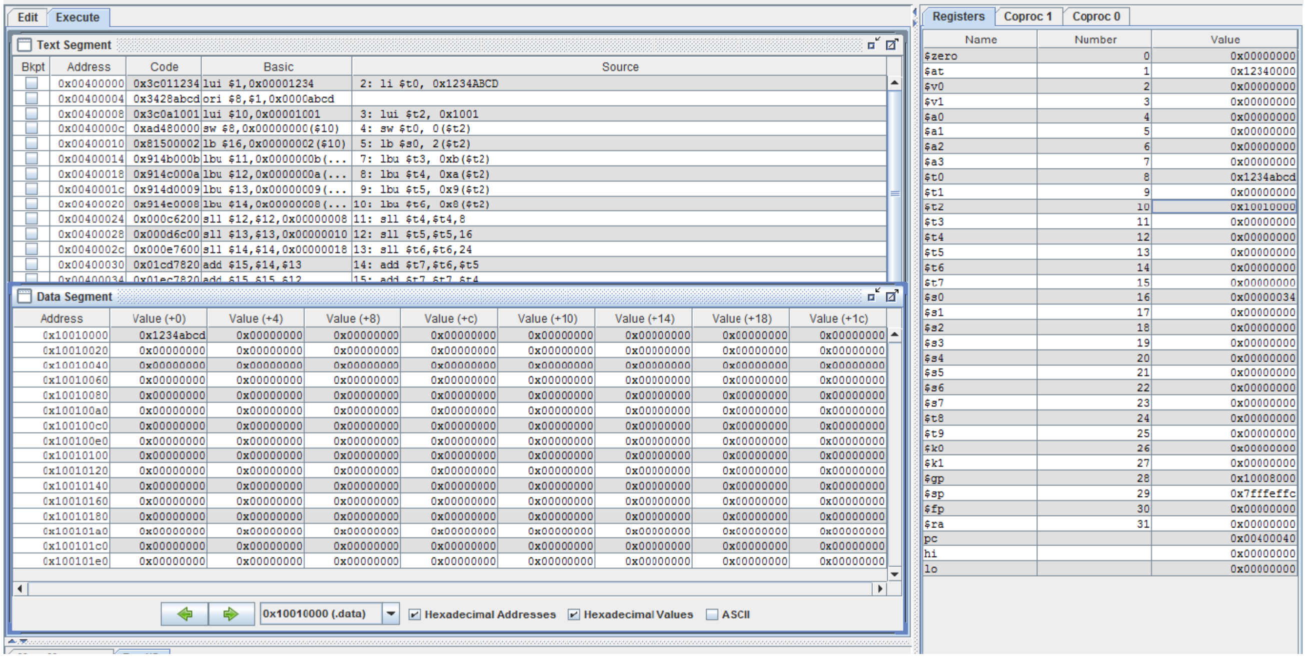
Name	Number	Value
$zero	0	0x00000000
$at	1	0x12340000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x1234abcd
$t1	9	0x00000000
$t2	10	0x10010000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000034
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040001c
hi		0x00000000
lo		0x00000000

4.3d)



```
#Question 4d
li $t0, 0x1234ABCD
lui $t2, 0x1001
sw $t0, 0($t2)
lb $s0, 2($t2)

lbu $t3, 0xb($t2)
lbu $t4, 0xa($t2)
lbu $t5, 0x9($t2)
lbu $t6, 0x8($t2)
sll $t4,$t4,8
sll $t5,$t5,16
sll $t6,$t6,24
add $t7,$t6,$t5
add $t7,$t7,$t4
add $t7,$t7,$t3
sw $t1,20($t2)
```



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x3c011234	lui $1,0x00001234	2: li $t0, 0x1234ABCD
□	0x00400004	0x3428abcd	ori $8,$1,0x0000abcd	
□	0x00400008	0x3c0a1001	lui $10,0x00001001	3: lui $t2, 0x1001
□	0x0040000c	0xad480000	sw $8,0x00000000($10)	4: sw $t0, 0($t2)
□	0x00400010	0x81500002	lb $16,0x00000002($10)	5: lb $s0, 2($t2)
□	0x00400014	0x914b000b	lbu $11,0x0000000b(...	7: lbu $t3, 0xb($t2)
□	0x00400018	0x914c000a	lbu $12,0x0000000a(...	8: lbu $t4, 0xa($t2)
□	0x0040001c	0x914d0009	lbu $13,0x00000009(...	9: lbu $t5, 0x9($t2)
□	0x00400020	0x914e0008	lbu $14,0x00000008(...	10: lbu $t6, 0x8($t2)
□	0x00400024	0x000c6200	sll $12,$12,0x00000008	11: sll $t4,$t4,8
□	0x00400028	0x000d6c00	sll $13,$13,0x00000010	12: sll $t5,$t5,16
□	0x0040002c	0x000e7600	sll $14,$14,0x00000018	13: sll $t6,$t6,24
□	0x00400030	0x01cd7820	add $15,$14,$13	14: add $t7,$t6,$t5
□	0x00400034	0x01ec7820	add $15,$15,$12	15: add $t7,$t7,$t4

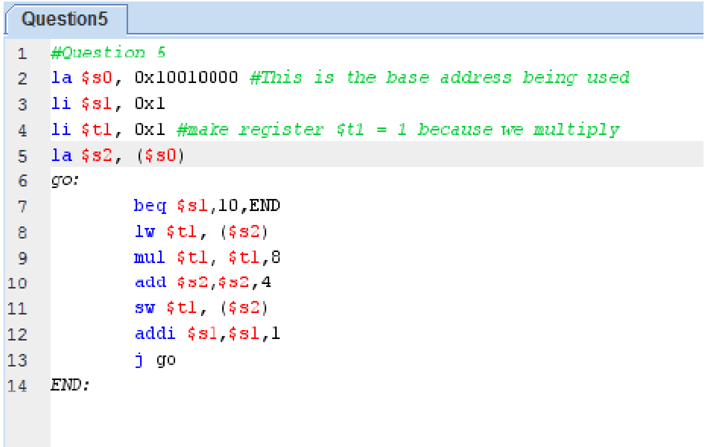
Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x1234abcd	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values □ ASCII

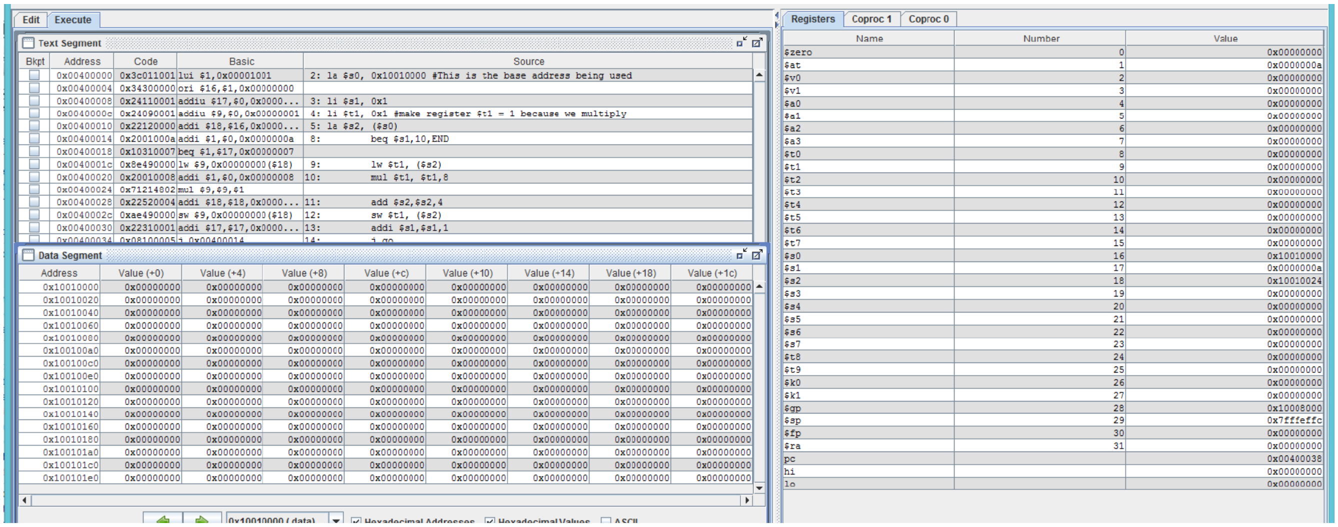
Name	Number	Value
$zero	0	0x00000000
$at	1	0x12340000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x1234abcd
$t1	9	0x00000000
$t2	10	0x10010000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000034
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400040
hi		0x00000000
lo		0x00000000

5)



Question5

```
#Question 5
la $s0, 0x10010000 #This is the base address being used
li $s1, 0x1
li $t1, 0x1 #make register $t1 = 1 because we multiply
la $s2, ($s0)
go:
          beq $s1,10,END
          lw $t1, ($s2)
          mul $t1, $t1,8
          add $s2,$s2,4
          sw $t1, ($s2)
          addi $s1,$s1,1
          j go
END:
```



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x3c011001	lui $1,0x00001001	2: la $s0, 0x10010000 #This is the base address being used
□	0x00400004	0x34300000	ori $16,$1,0x00000000	
□	0x00400008	0x24110001	addiu $17,$0,0x0000...	3: li $s1, 0x1
□	0x0040000c	0x24090001	addiu $9,$0,0x00000001	4: li $t1, 0x1 #make register $t1 = 1 because we multiply
□	0x00400010	0x22120000	addi $18,$16,0x0000...	5: la $s2, ($s0)
□	0x00400014	0x2001000a	addi $1,$0,0x0000000a	8: beq $s1,10,END
□	0x00400018	0x10310007	beq $1,$17,0x00000007	
□	0x0040001c	0x8e490000	lw $9,0x00000000($18)	9: lw $t1, ($s2)
□	0x00400020	0x20010008	addi $1,$0,0x00000008	10: mul $t1, $t1,8
□	0x00400024	0x71214802	mul $9,$9,$1	
□	0x00400028	0x22520004	addi $18,$18,0x0000...	11: add $s2,$s2,4
□	0x0040002c	0xae490000	sw $9,0x00000000($18)	12: sw $t1, ($s2)
□	0x00400030	0x22310001	addi $17,$17,0x0000...	13: addi $s1,$s1,1
□	0x00400034	0x08100005	j 0x00400014	14: j go

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0x00000000
$at	1	0x0000000a
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x10010000
$s1	17	0x0000000a
$s2	18	0x10010024
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400038
hi		0x00000000
lo		0x00000000

6)

Regwrite: 1

RegDest: 1

ALUSrc: 01

Branch: 0

MemWrite: 0

MemtoReg: 0

ALUControl: 00 (whichever 2-bit binary makes the ALU add)