CCCN 221 – Computer Architecture

LAB#5 Task4

Task Date: As per BB **Submission Date: As per BB**

Student Name: Amin Yahya Selhabi **Student ID:** 2140632

Note: Student must attach the code and the screenshot of the Final output using MIPS or Qtsmpm.

Marks:

Exercises	1	2	3	Total
Allocated	1	1	1	3
Obtained				
CLO, PLO, SO	3.1, V3, S05	3.1, V3, S05	3.1, V3, S05	

1. Task

Implement the following assembly program in Mars MIPS that performs the following tasks:

Program to find the largest of 3 numbers. *Read Lab5 sheet.*

screenshot of the Final output using MIPS or Qtsmpm

Solution at the end of this file

Output Sample.

```
Enter first number: 15
Enter second number: 3
Enter third number: 20
20 is the greatest number
-- program is finished running --
```

2. Task

Implement the following assembly program in Mars MIPS that performs the following tasks:

Program to find the smallest of 3 numbers. *Read Lab5 sheet.*

screenshot of the Final output using MIPS or Qtsmpm

Solution at the end of this file

Output Sample.

```
Enter first number: 15
Enter second number: 3
Enter third number: 20
3 is the smallest number is
-- program is finished running --
```

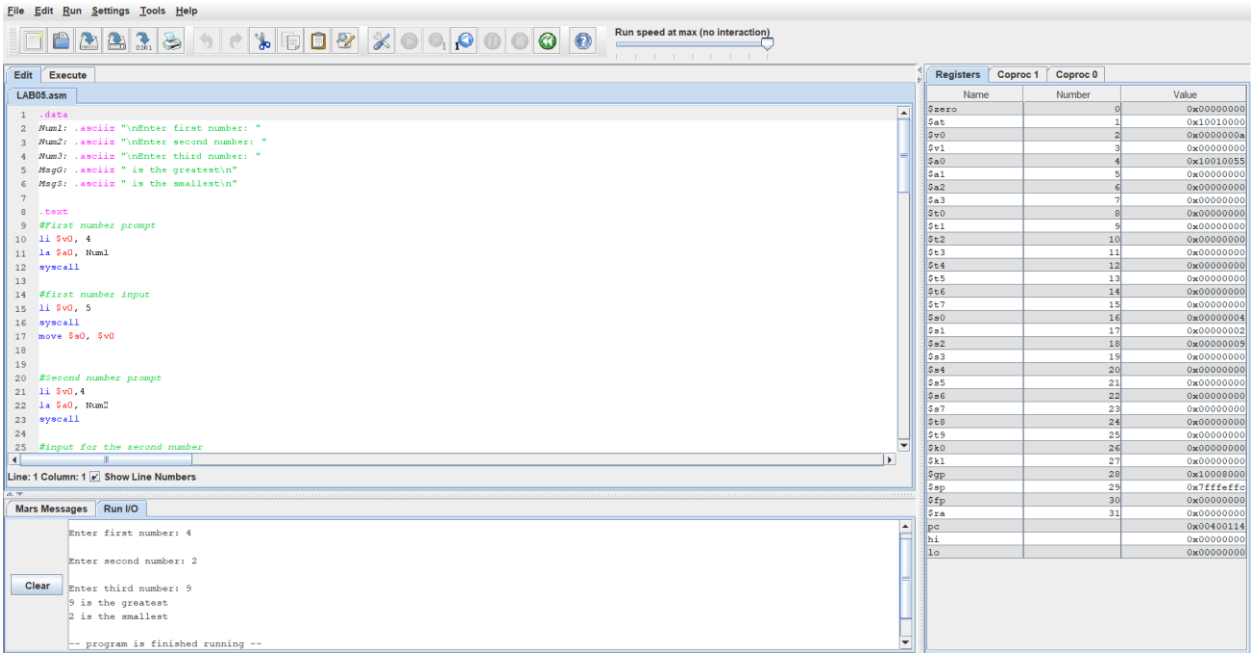
3. **Answer the Question with the Reason without explanation, answered will be not considered.** *Read Lab5 sheet.*

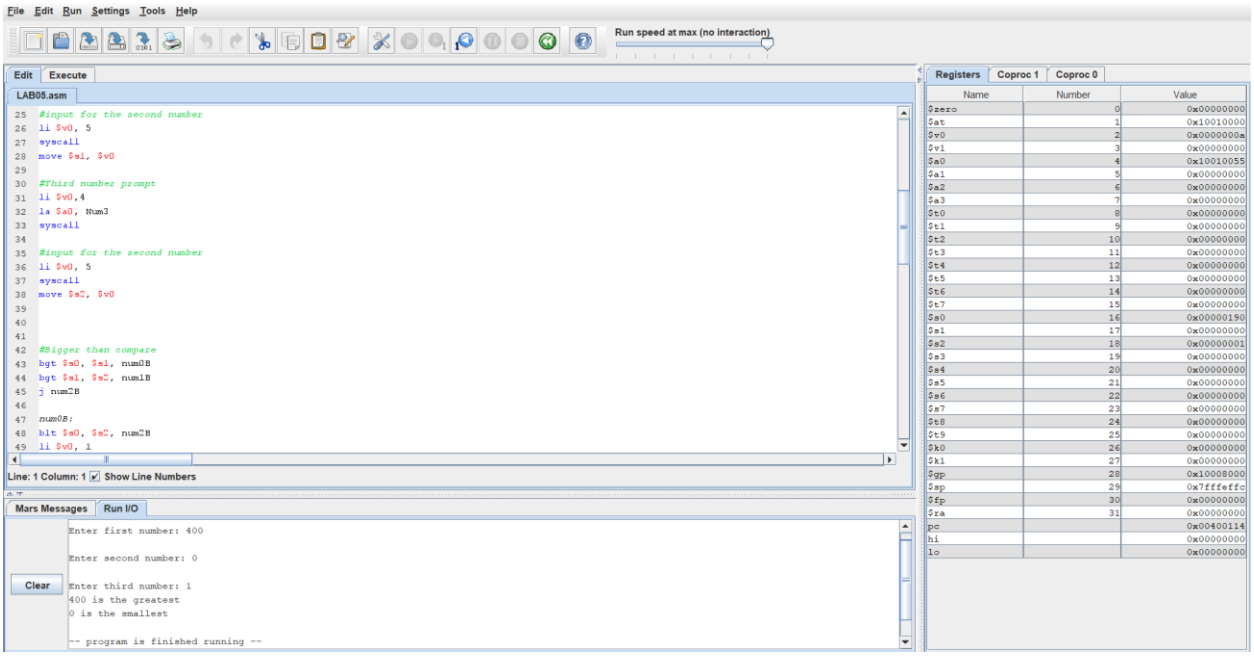
To what address will the following instruction jump when the branch is taken? The instruction is at address 12.

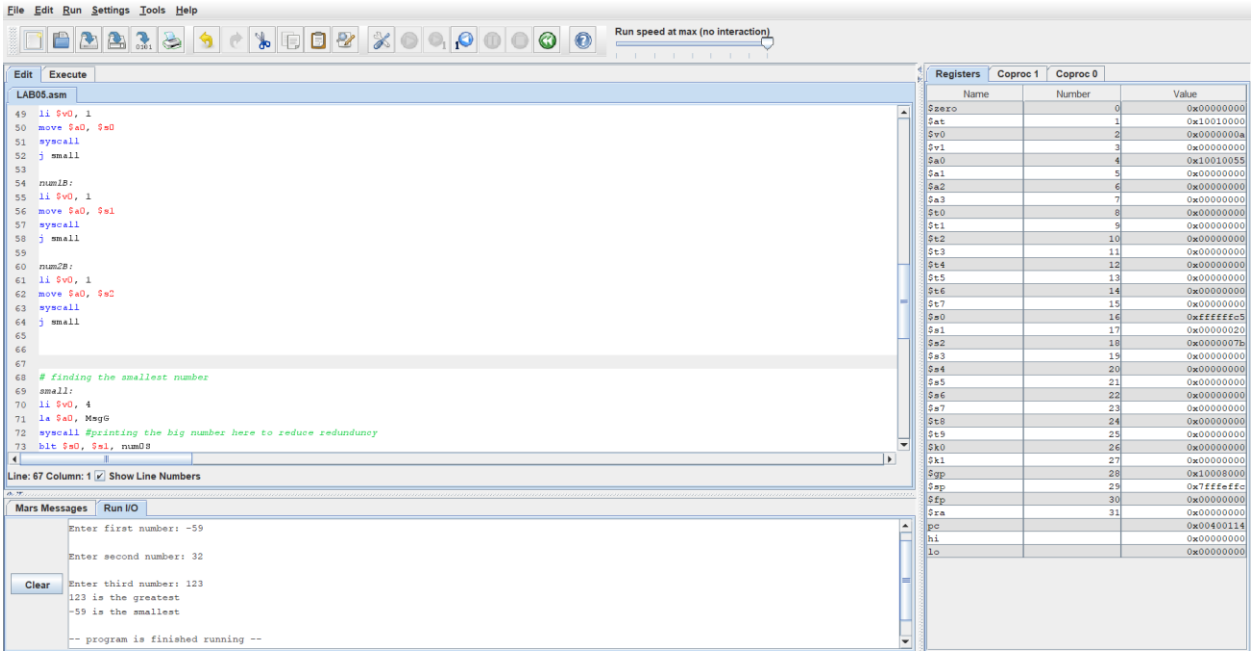
a. 76
b. 32
c. 80
d. 28

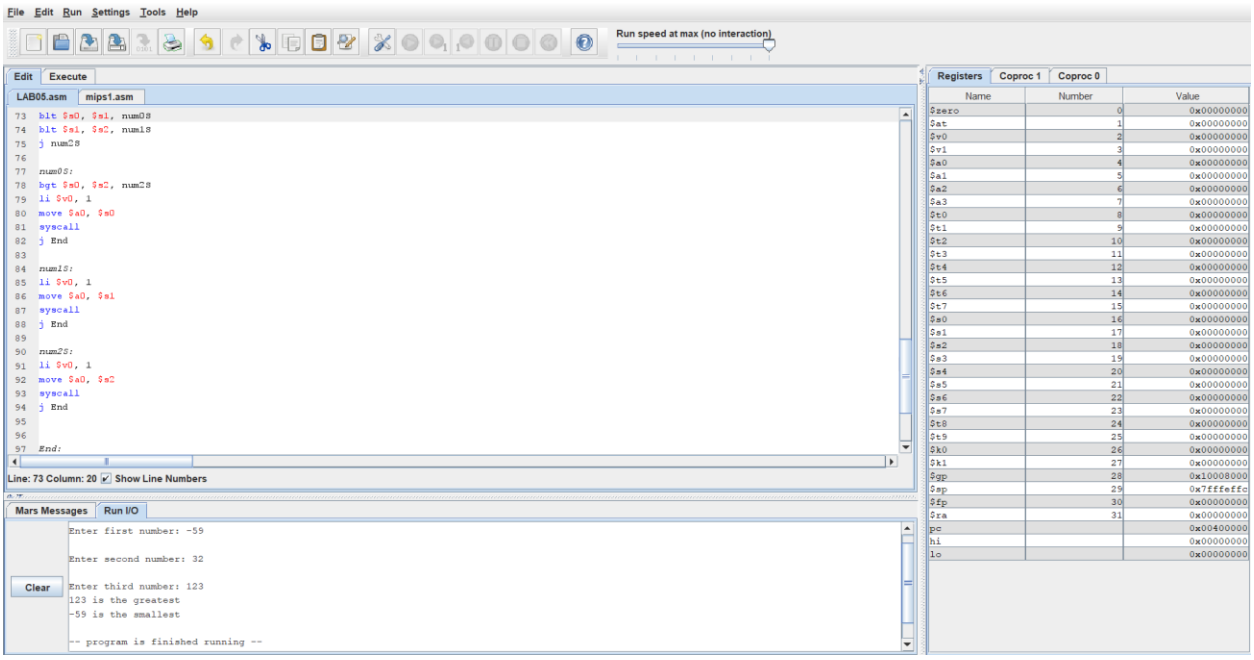
Because the branch will jump to the current PC in addition to the immediate 16 bits times 4 then finally, we add 2 << 2 which is 4

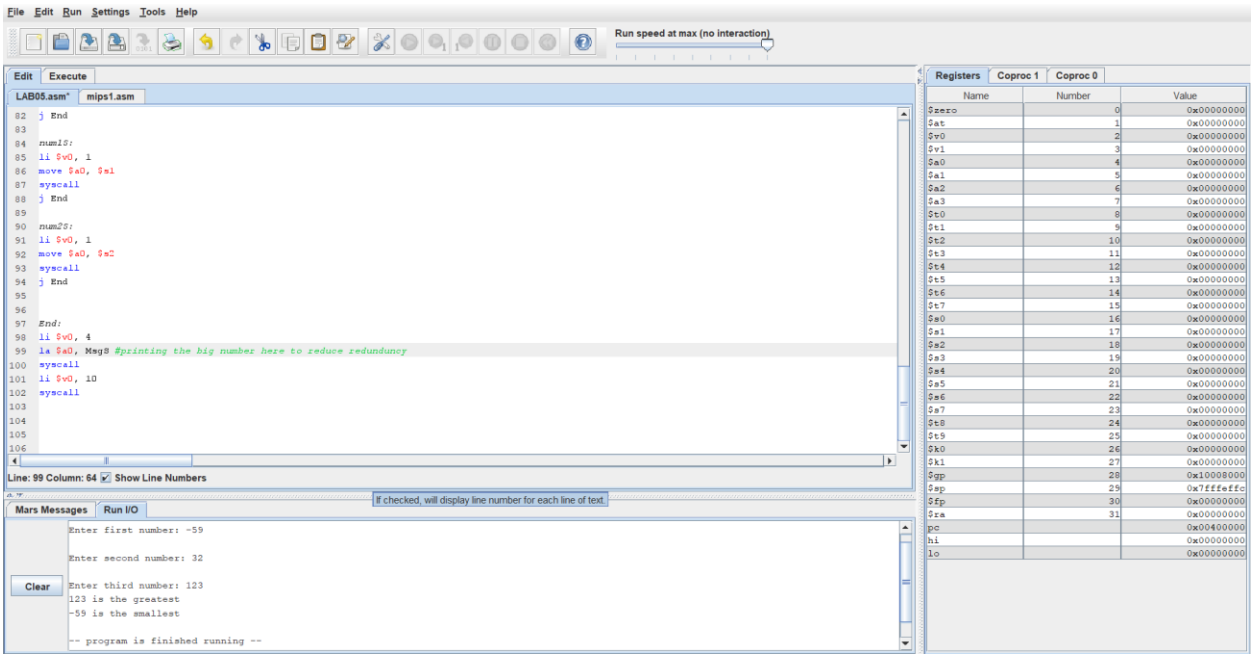
I have solved task 1 and task 2 on a single programme


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
LAB05.asm
1 .data
2 Num1: .asciiz "\nEnter first number: "
3 Num2: .asciiz "\nEnter second number: "
4 Num3: .asciiz "\nEnter third number: "
5 MsgG: .asciiz " is the greatest\n"
6 MsgS: .asciiz " is the smallest\n"
7
8 .text
9 #first number prompt
10 li $v0, 4
11 la $a0, Num1
12 syscall
13
14 #first number input
15 li $v0, 5
16 syscall
17 move $s0, $v0
18
19
20 #Second number prompt
21 li $v0,4
22 la $a0, Num2
23 syscall
24
25 #input for the second number
Line: 1 Column: 1 Show Line Numbers
Mars Messages Run I/O
Clear
Enter first number: 4
Enter second number: 2
Enter third number: 9
9 is the greatest
2 is the smallest
-- program is finished running --
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x10010000
$v0 2 0x0000000a
$v1 3 0x00000000
$a0 4 0x10010055
$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 0x00000004
$s1 17 0x00000002
$s2 18 0x00000009
$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 0x00400114
hi 0x00000000
lo 0x00000000


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
LAB05.asm
25 #input for the second number
26 li $v0, 5
27 syscall
28 move $s1, $v0
29
30 #Third number prompt
31 li $v0,4
32 la $a0, Num3
33 syscall
34
35 #input for the second number
36 li $v0, 5
37 syscall
38 move $s2, $v0
39
40
41
42 #Bigger than compare
43 bgt $s0, $s1, num0B
44 bgt $s1, $s2, num1B
45 j num2B
46
47 num0B:
48 blt $s0, $s2, num2B
49 li $v0, 1
Line: 1 Column: 1 Show Line Numbers
Mars Messages Run I/O
Clear
Enter first number: 400
Enter second number: 0
Enter third number: 1
400 is the greatest
0 is the smallest
-- program is finished running --
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x10010000
$v0 2 0x0000000a
$v1 3 0x00000000
$a0 4 0x10010055
$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 0x00000190
$s1 17 0x00000000
$s2 18 0x00000001
$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 0x00400114
hi 0x00000000
lo 0x00000000


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
LAB05.asm
49 li $v0, 1
50 move $a0, $m0
51 syscall
52 j small
53
54 numlB:
55 li $v0, 1
56 move $a0, $s1
57 syscall
58 j small
59
60 num2B:
61 li $v0, 1
62 move $a0, $s2
63 syscall
64 j small
65
66
67
68 # finding the smallest number
69 small:
70 li $v0, 4
71 la $a0, MsgG
72 syscall #printing the big number here to reduce redundancy
73 blt $m0, $s1, num0S
Line: 67 Column: 1 Show Line Numbers
Mars Messages Run I/O
Clear
Enter first number: -59
Enter second number: 32
Enter third number: 123
123 is the greatest
-59 is the smallest
-- program is finished running --
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x10010000
$v0 2 0x0000000a
$v1 3 0x00000000
$a0 4 0x10010055
$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 0xfffffffc5
$s1 17 0x00000020
$s2 18 0x0000007b
$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 0x00400114
hi 0x00000000
lo 0x00000000


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
LAB05.asm mips1.asm
73 blt $m0, $s1, num0S
74 blt $s1, $s2, num1S
75 j num2S
76
77 num0S:
78 bgt $m0, $s2, num2S
79 li $v0, 1
80 move $a0, $m0
81 syscall
82 j End
83
84 num1S:
85 li $v0, 1
86 move $a0, $s1
87 syscall
88 j End
89
90 num2S:
91 li $v0, 1
92 move $a0, $s2
93 syscall
94 j End
95
96
97 End:
Line: 73 Column: 20 Show Line Numbers
Mars Messages Run I/O
Clear
Enter first number: -59
Enter second number: 32
Enter third number: 123
123 is the greatest
-59 is the smallest
-- program is finished running --
Registers Coproc 1 Coproc 0
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 5 0x00000000
$t2 10 0x00000000
$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 0x00400000
hi 0x00000000
lo 0x00000000



Code:

```
.data
Num1: .asciiz "\nEnter first number: "
Num2: .asciiz "\nEnter second number: "
Num3: .asciiz "\nEnter third number: "
MsgG: .asciiz " is the greatest\n"
MsgS: .asciiz " is the smallest\n"


.text
#First number prompt
li $v0, 4
la $a0, Num1
syscall
```

```
#first number input
li $v0, 5
syscall
move $s0, $v0


#Second number prompt
li $v0,4
la $a0, Num2
syscall

#input for the second number
li $v0, 5
syscall
move $s1, $v0

#Third number prompt
li $v0,4
la $a0, Num3
syscall

#input for the second number
li $v0, 5
syscall
```

```
move $s2, $v0



#Bigger than compare
bgt $s0, $s1, num0B
bgt $s1, $s2, num1B
j num2B

num0B:
blt $s0, $s2, num2B
li $v0, 1
move $a0, $s0
syscall
j small

num1B:
li $v0, 1
move $a0, $s1
syscall
j small

num2B:
li $v0, 1
```

```
move $a0, $s2
syscall
j small



# finding the smallest number
small:
li $v0, 4
la $a0, MsgG
syscall #printing the big number here to reduce redunduncy
blt $s0, $s1, num0S
blt $s1, $s2, num1S
j num2S

num0S:
bgt $s0, $s2, num2S
li $v0, 1
move $a0, $s0
syscall
j End

num1S:
li $v0, 1
```

```
move $a0, $s1
syscall
j End

num2S:
li $v0, 1
move $a0, $s2
syscall
j End


End:
li $v0, 4
la $a0, MsgS #printing the big number here to reduce redunduncy
syscall
li $v0, 10
syscall
```