**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 -- |

1. **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 -- |

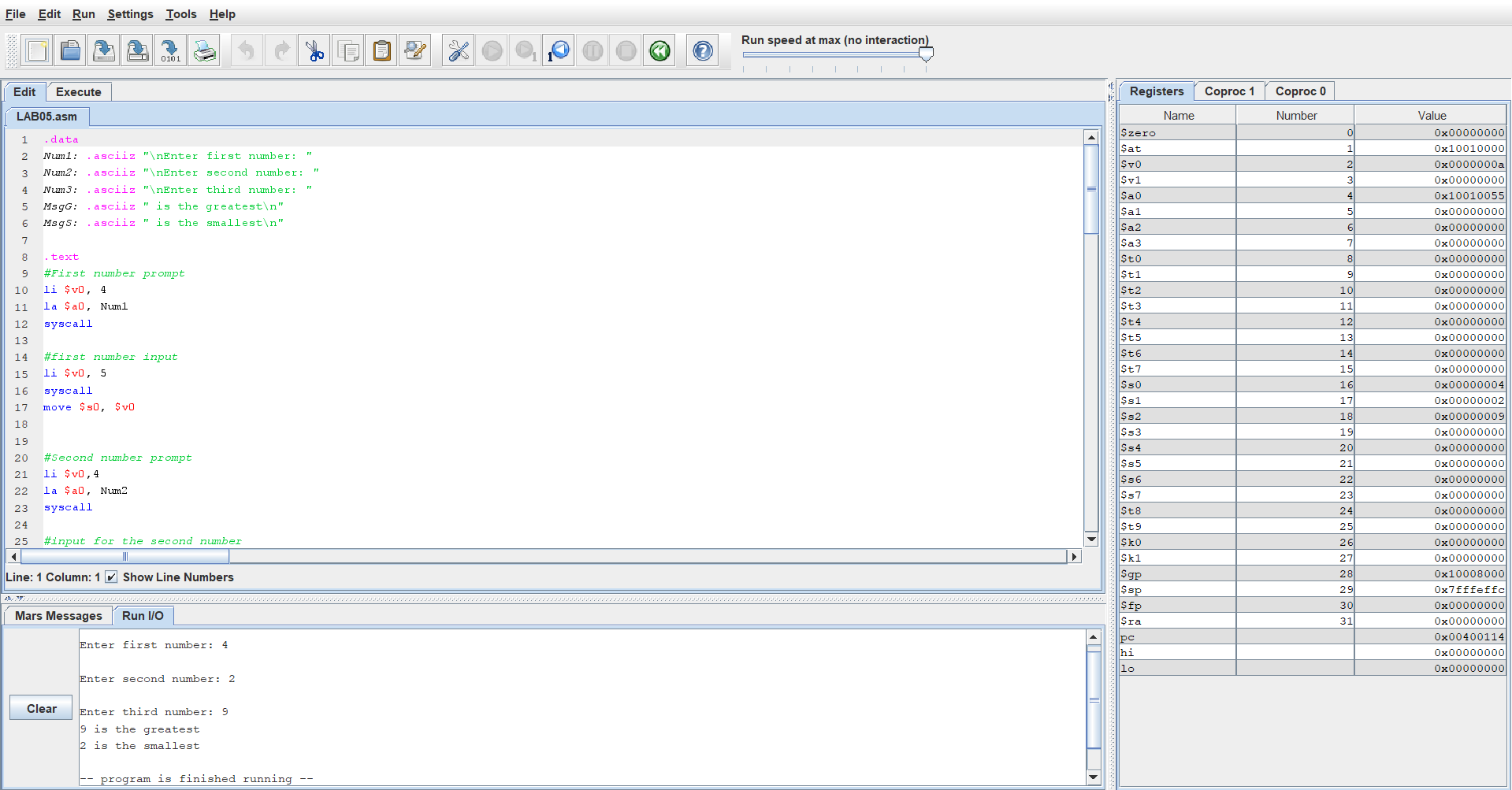
1. **Answer the Question with the Reason without explanation, answered will be not considered.** *Read Lab5 sheet.*

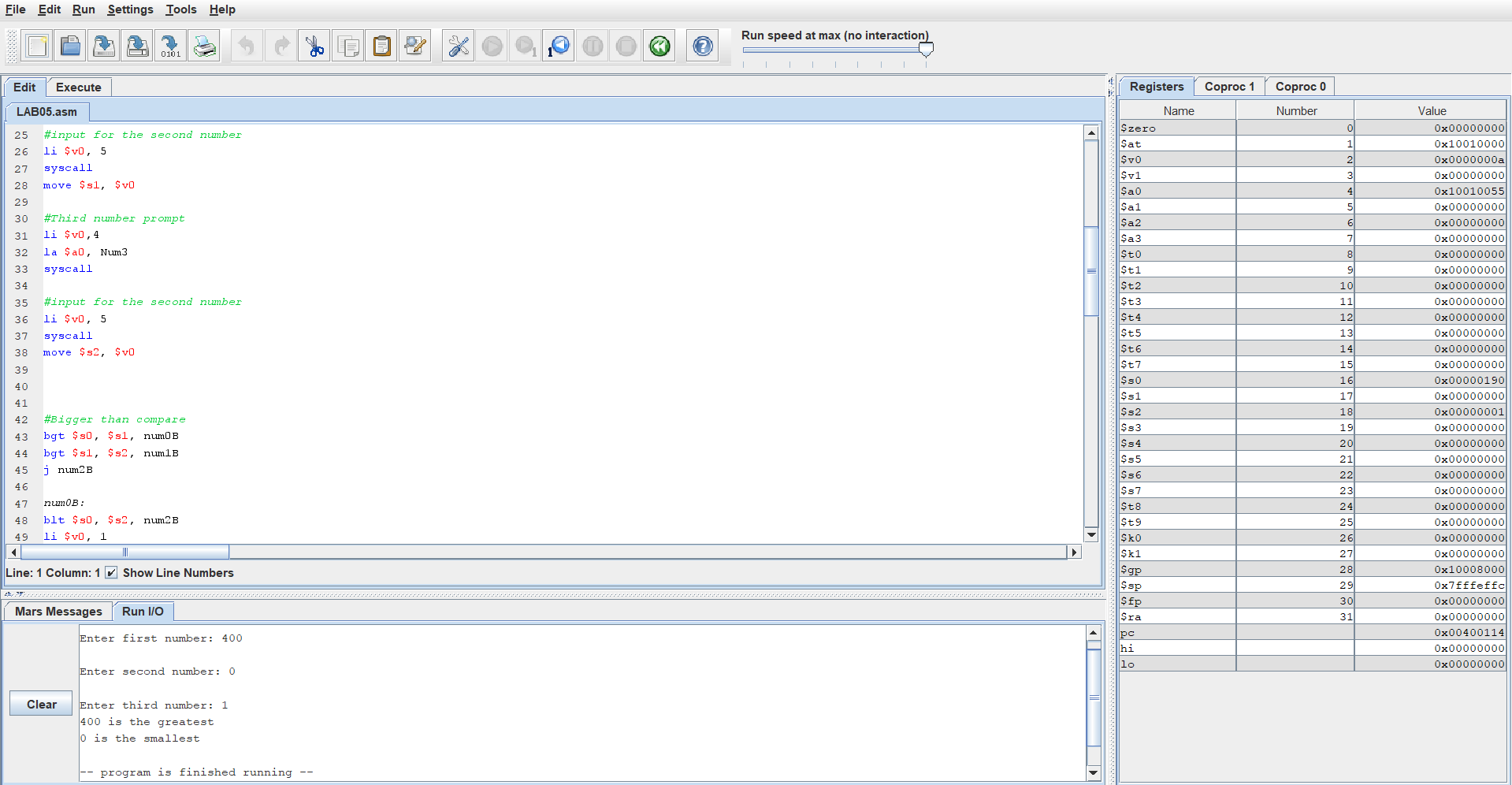
***To what address will the following instruction jump when the branch is taken? The instruction is at address 12.***

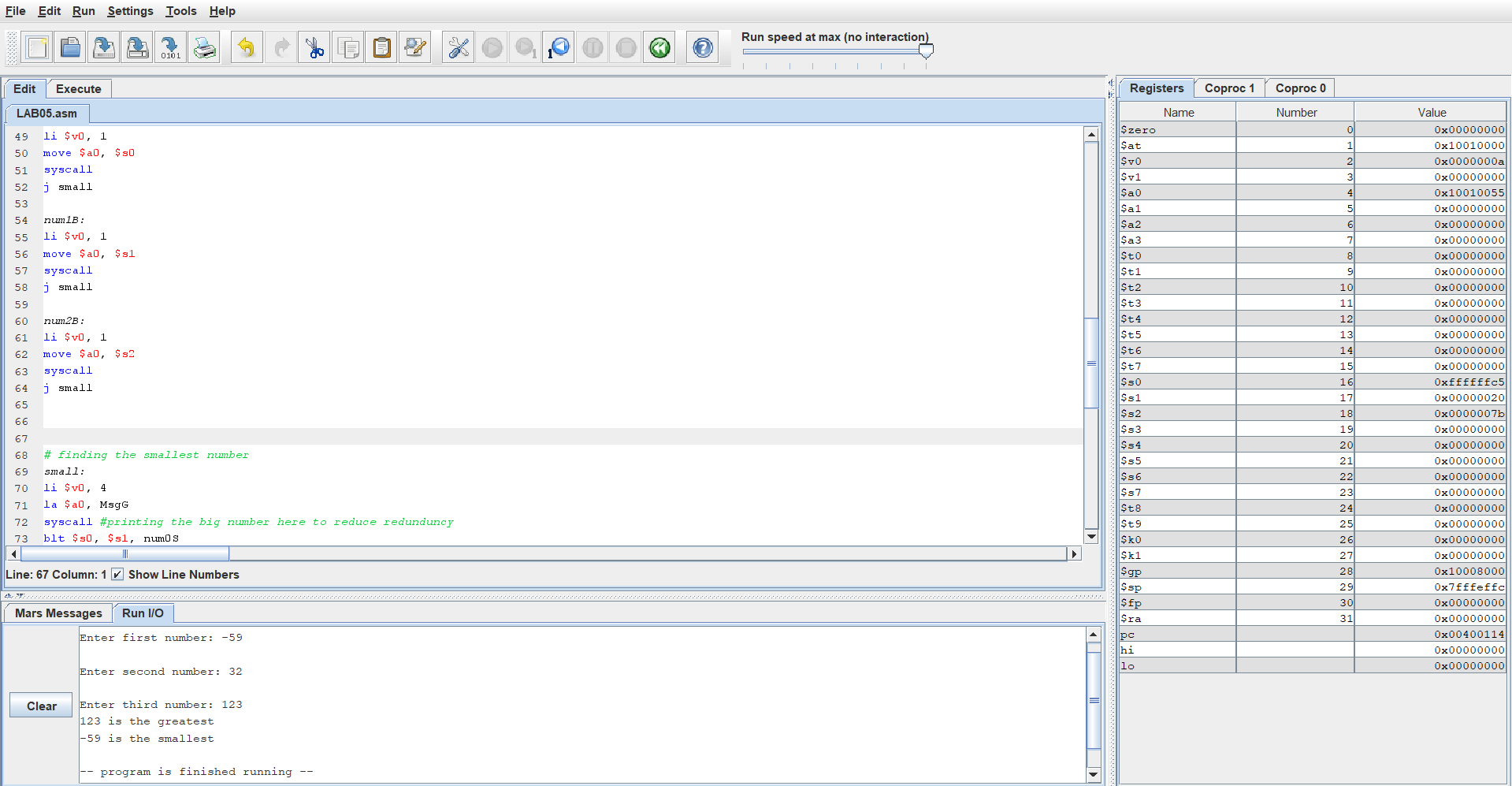
1. 76
2. 32
3. 80
4. 28

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







Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application

Description automatically generated

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