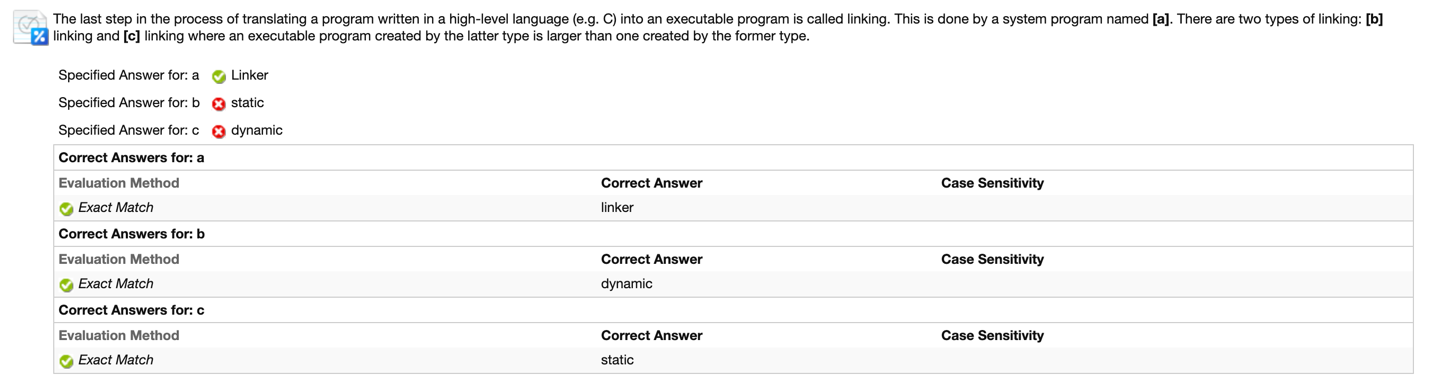
Exam II Rework

Q1:



Cause of Error:

Carelessness. Did not focus on the latter explanation.

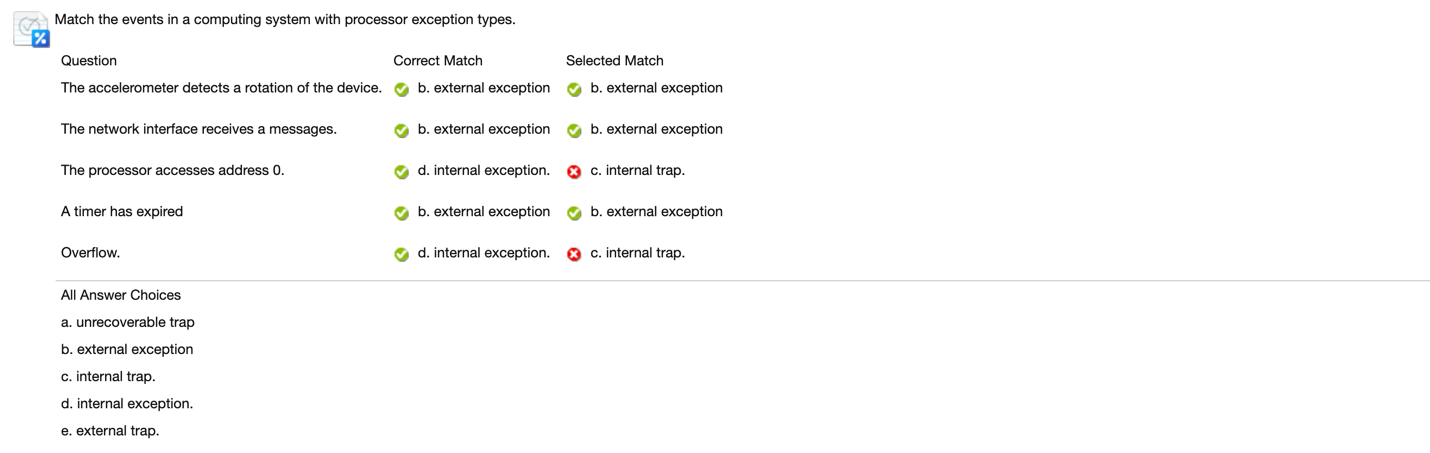
Rework Solution:

B: dynamic

C: static

The executable program created by dynamic linking is smaller than the one by static linking.

Q2:



Cause of Error:

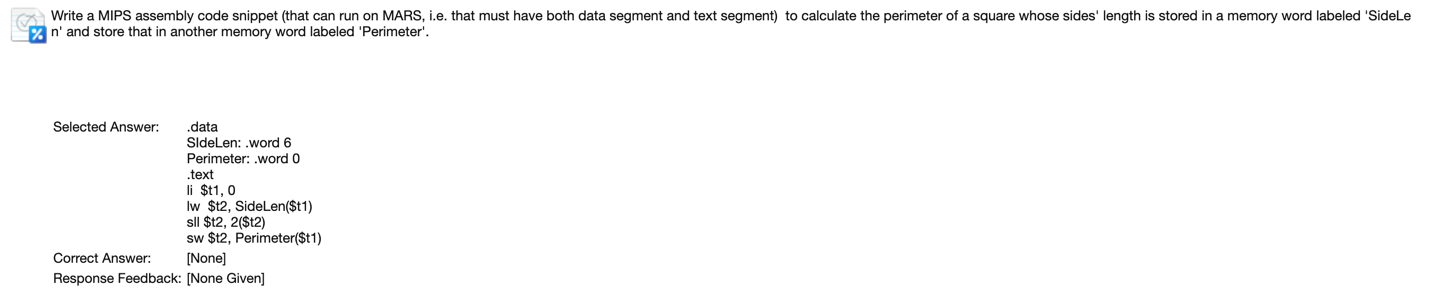
Chaos in my memory. Internal exception is trap, but internal trap is nothing.

Rework Solution:

3) D. Internal exception

5) D. Internal exception

Q6:



Cause of Error:

I was running out of time and what I could do that time is using the example of MyArray to fill the blank.

Rework Solution:

.data

SideLen: .word 8

Perimeter: .space 4

.text

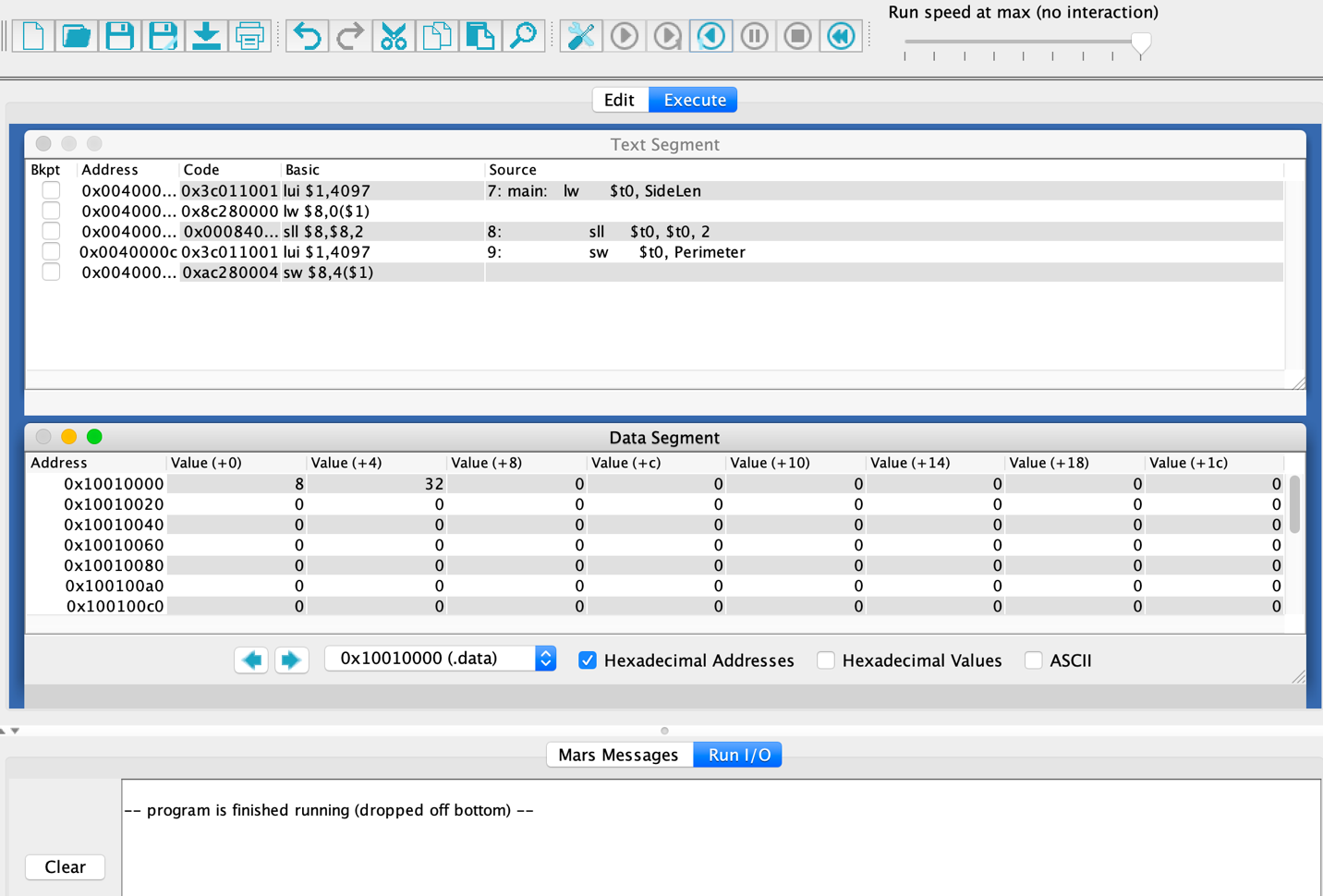
.globl main

main: lw $t0, SideLen

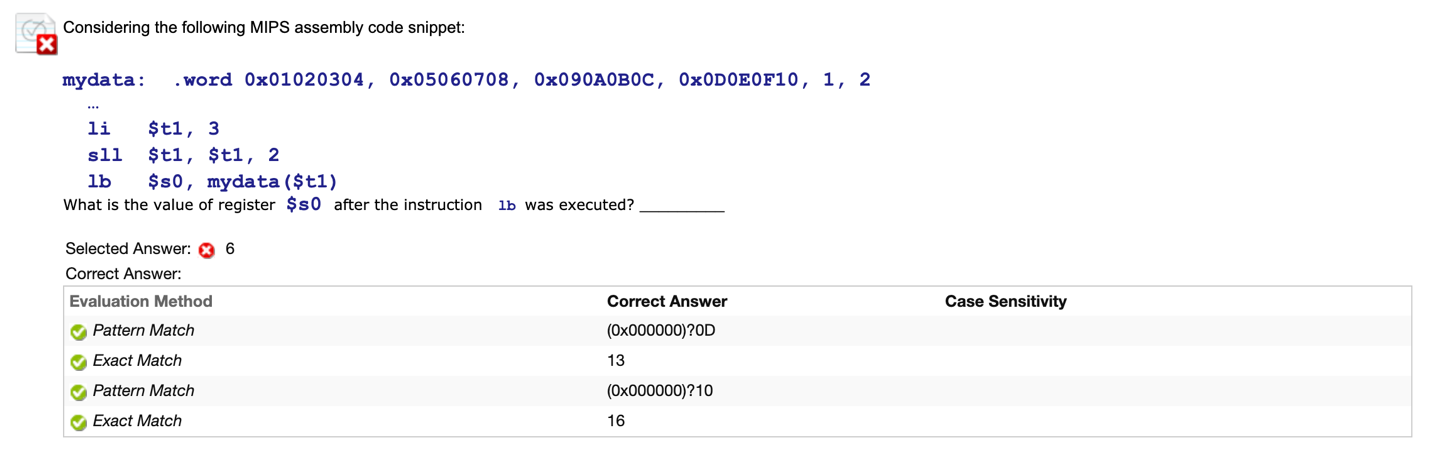
sll $t0, $t0, 2

sw $t0, Perimeter

Check on MarsPlus:



Q7:



Cause of Error:

I found out that this question use lb, but I misremembered the definition of it.

Rework Solution:

$t1 = 3 \* 4 = 12, So this code is equals to get byte at mydata[12].

Memory:

00000002 20

00000001 16

0D0E0F10 12

090A0B0C 8

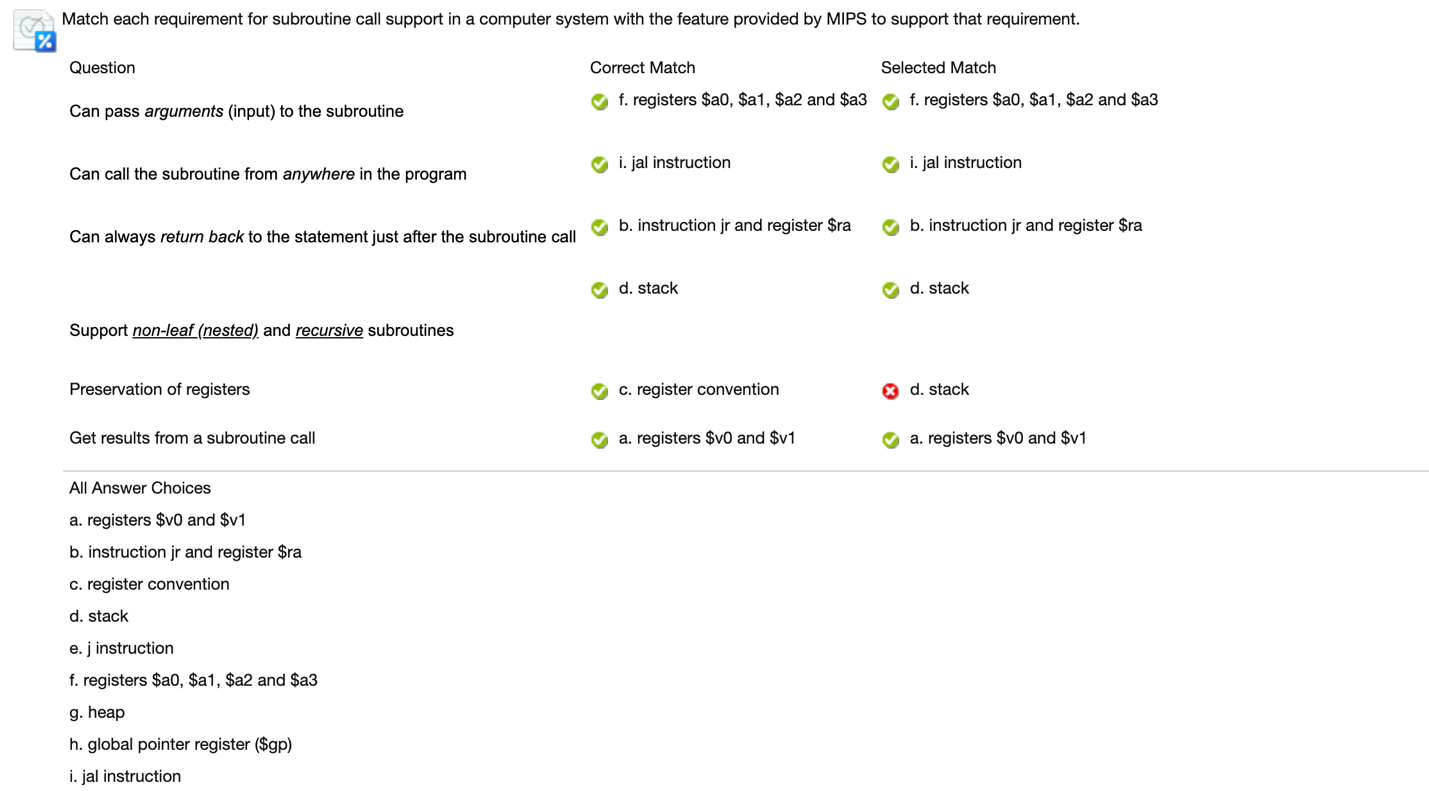
05060708 4

01020304 0

Little-Endian: 0x10 = 16

Big-Endian: 0x0D = 13

Q8:



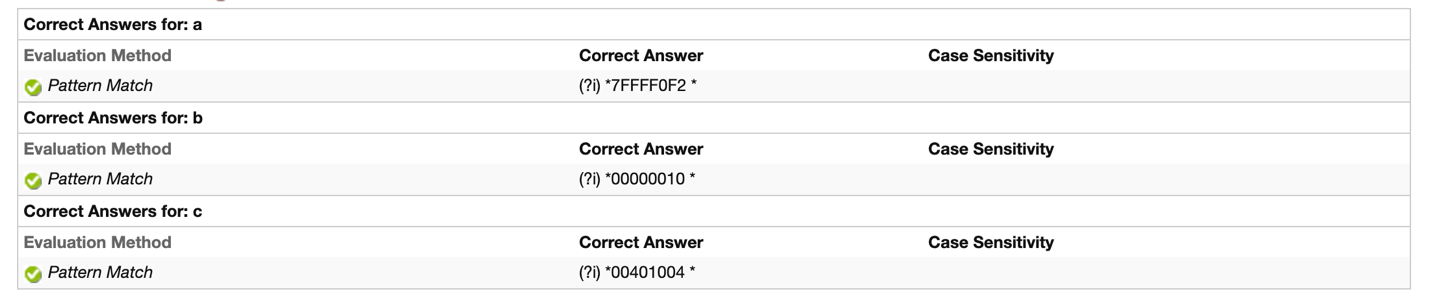
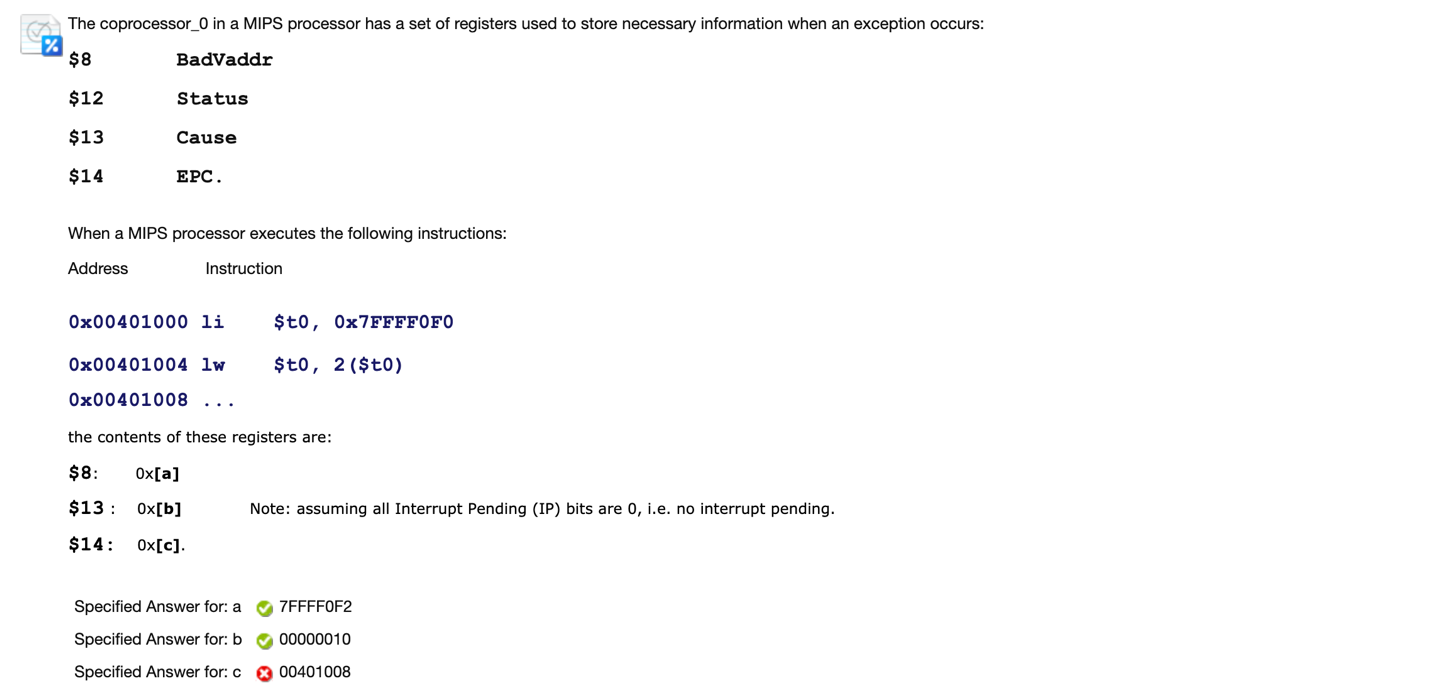
Cause of Error:

I slipped to a wrong answer.

Rework Solution:

Choose C. Registers are preserved in register convention in MIPS.

Q15:



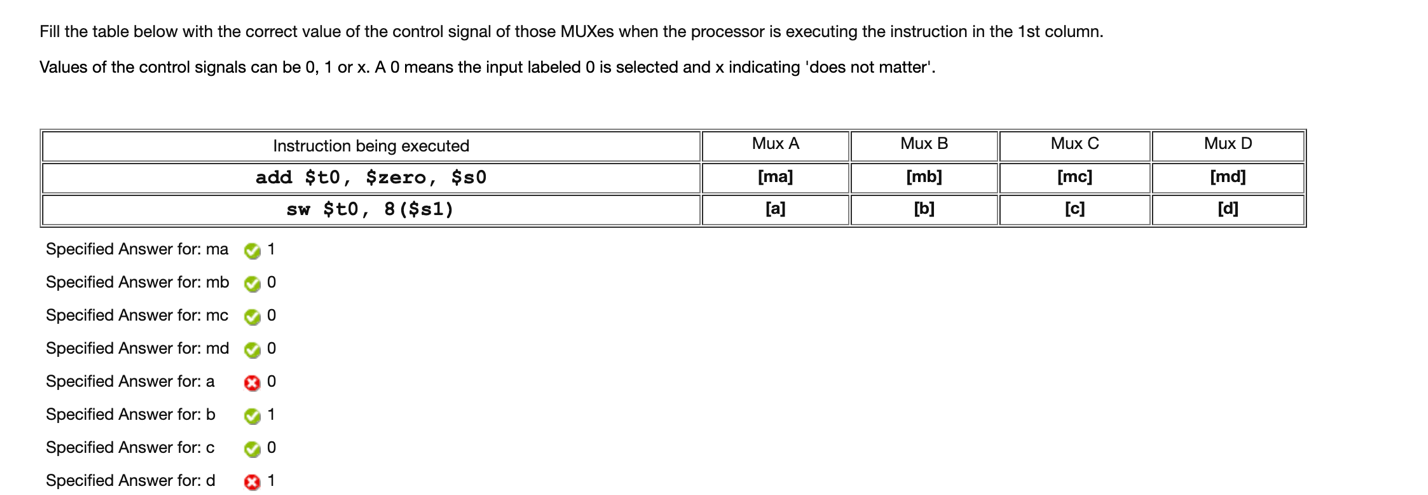
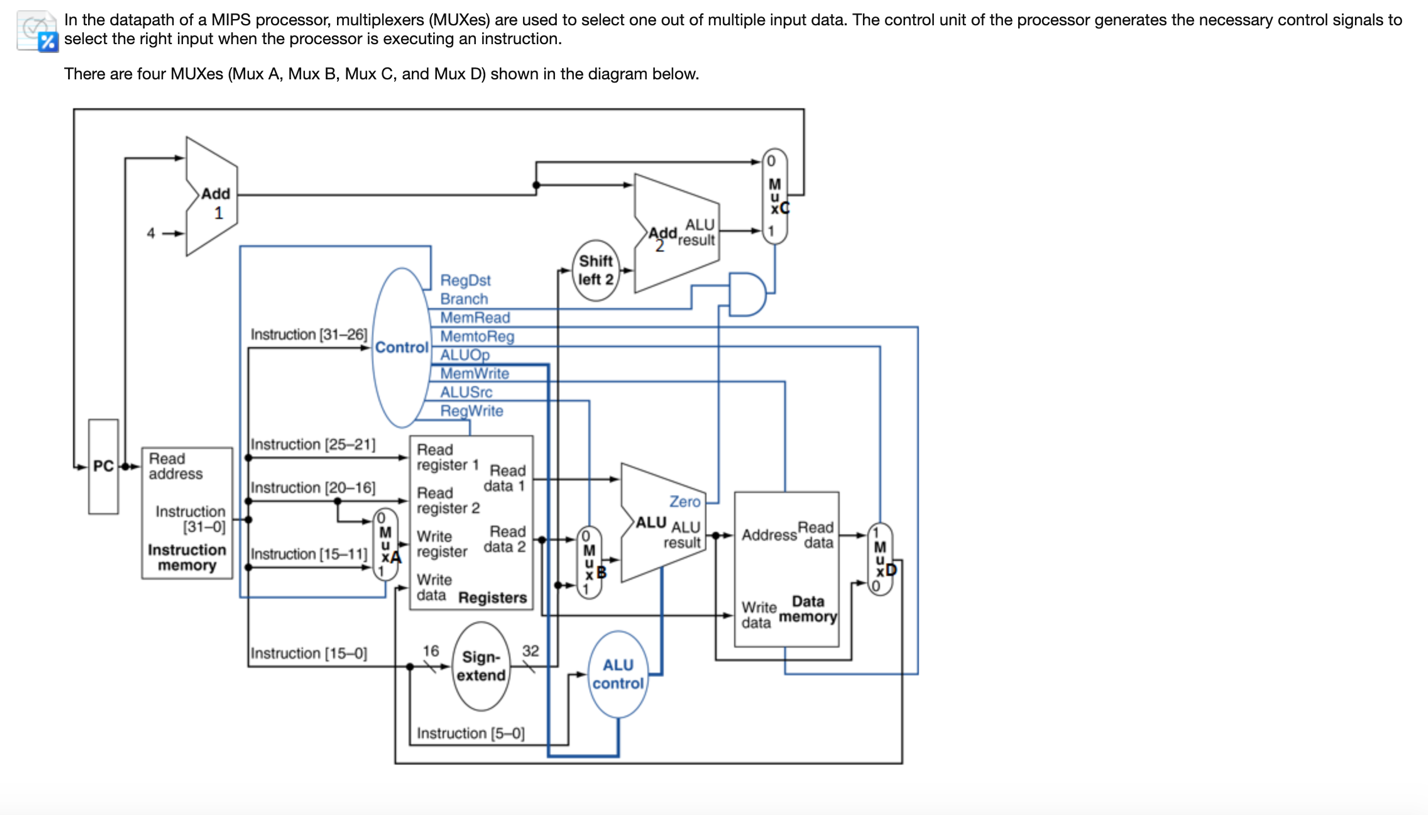
Cause of Error:

Hardware copies PC – 4 into EPC($14) on coprocessor\_0

Rework Solution:

EPC = 0x00401008 – 4 = 0x00401004

Q22:



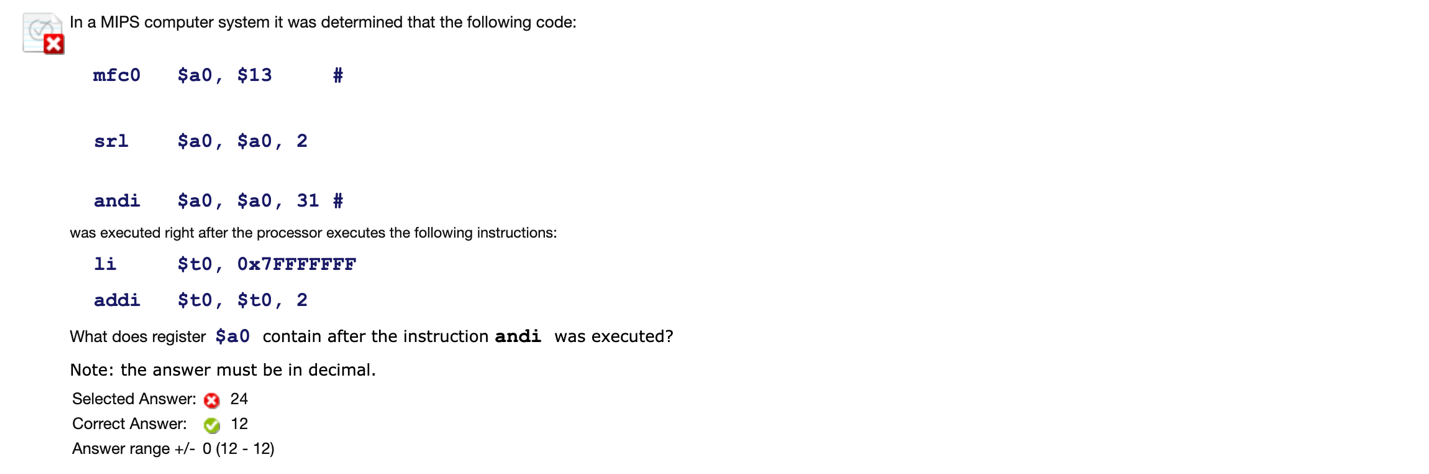
Cause of Error:

I was short of time at the end of this exam and I did not see the option for x. So I just give the answer of 0 and 1 even I found something weird.

Rework Solution:

When sw, we do not use mux A and D, so blank a and d is x.

Q23:



Cause of Error:

Error in right shift…

Rework Solution:

Addi is addition for signed integer. Add 0x00000002 to 0x7FFFFFFF will cause overflow.

Cause error: Arithmetic Overflow Exception (Exception Code 12)

12 = 01100 31 = 00011111

$13 : XXXX XXXX XXXX XXXX XXXX XXXX X011 00XX

>>2: XXXX XXXX XXXX XXXX XXXX XXXX XXX0 1100

And 00011111 (get last five digit): $t0 = 01100 = 12