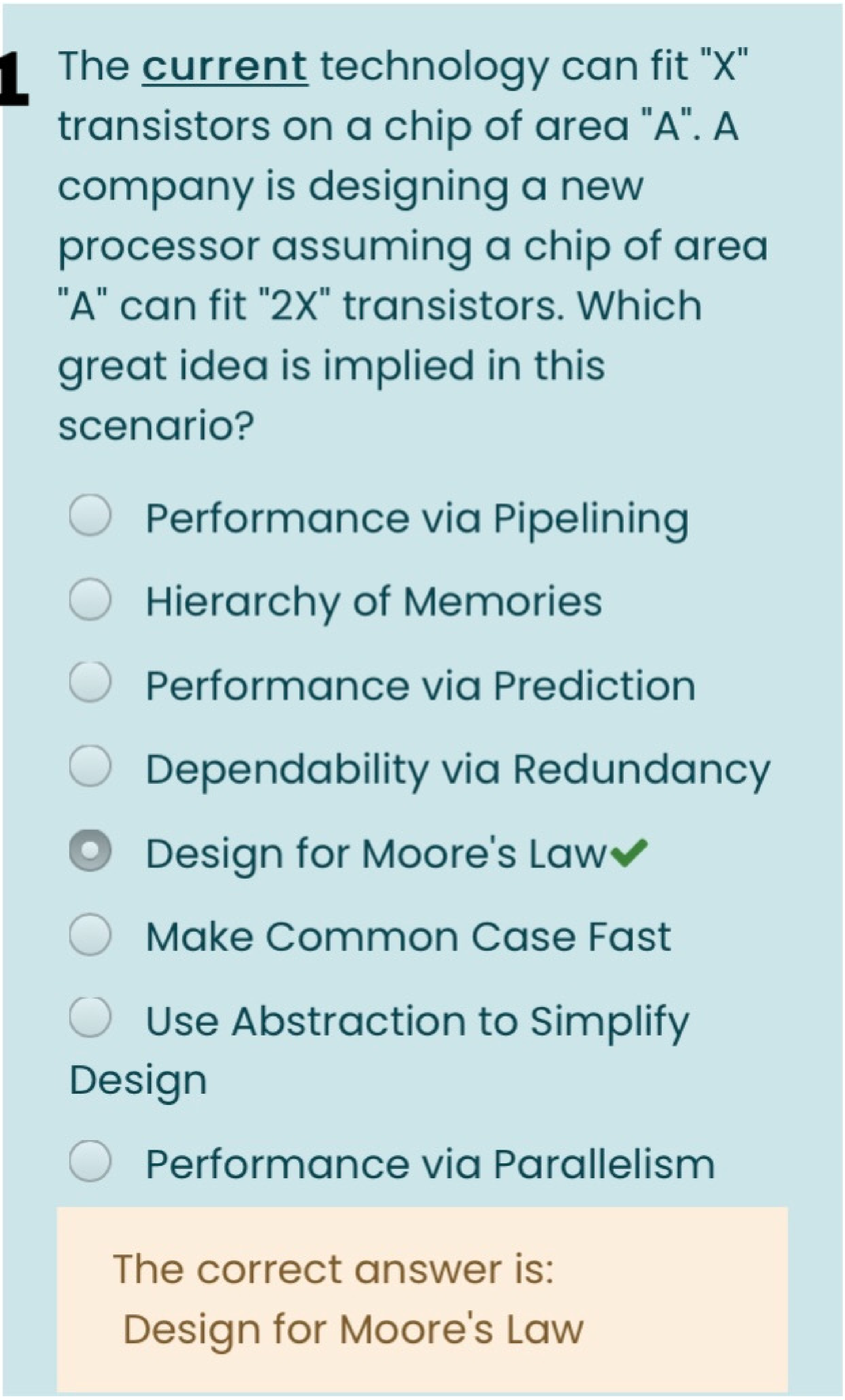
Q1



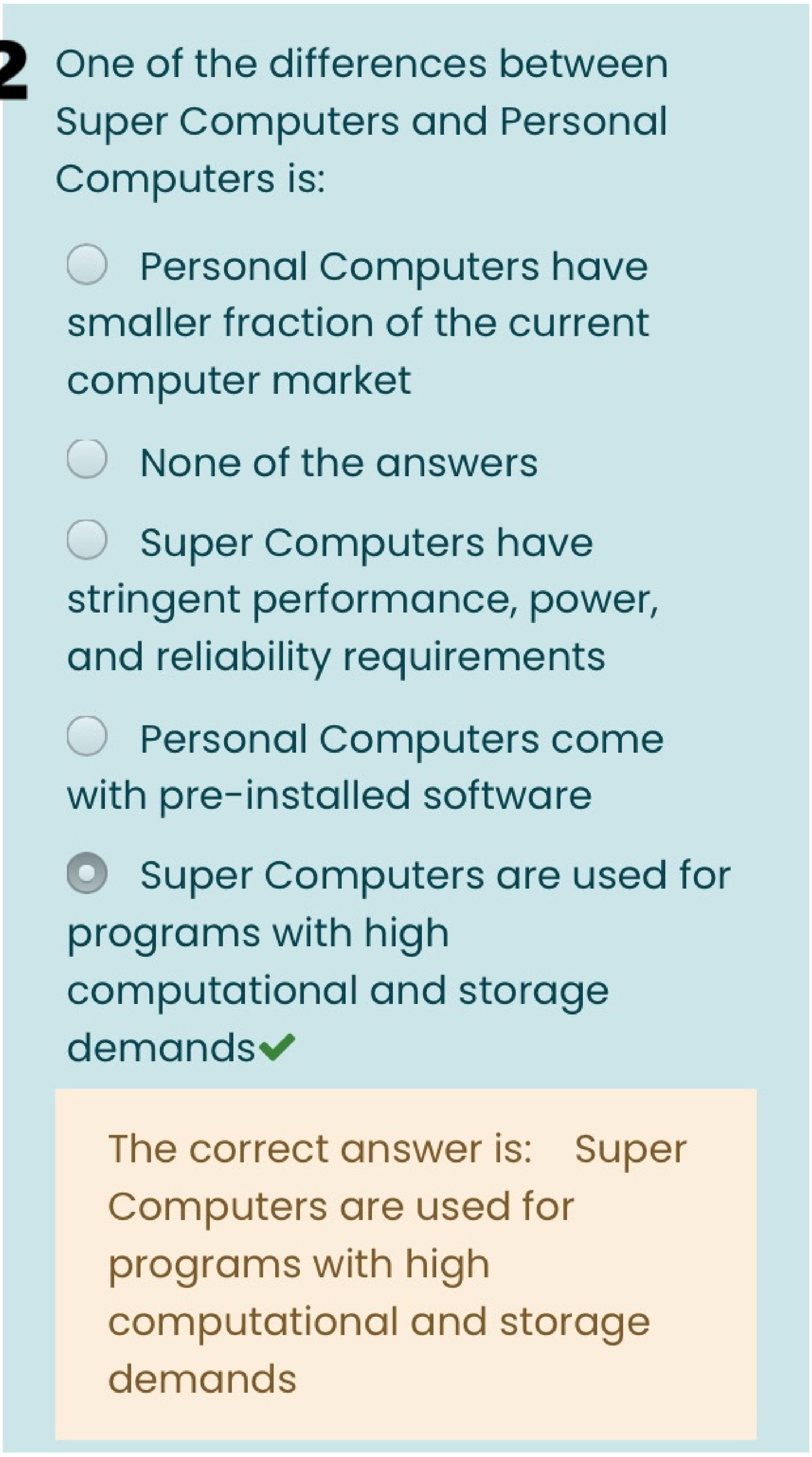
The **current** technology can fit "X" transistors on a chip of area "A". A company is designing a new processor assuming a chip of area "A" can fit "2X" transistors. Which great idea is implied in this scenario?

- Performance via Pipelining
- Hierarchy of Memories
- Performance via Prediction
- Dependability via Redundancy
- [✓ Design for Moore's Law] ✓
- Make Common Case Fast
- Use Abstraction to Simplify Design
- Performance via Parallelism

The correct answer is:
Design for Moore's Law

POWERUNIT

Q2

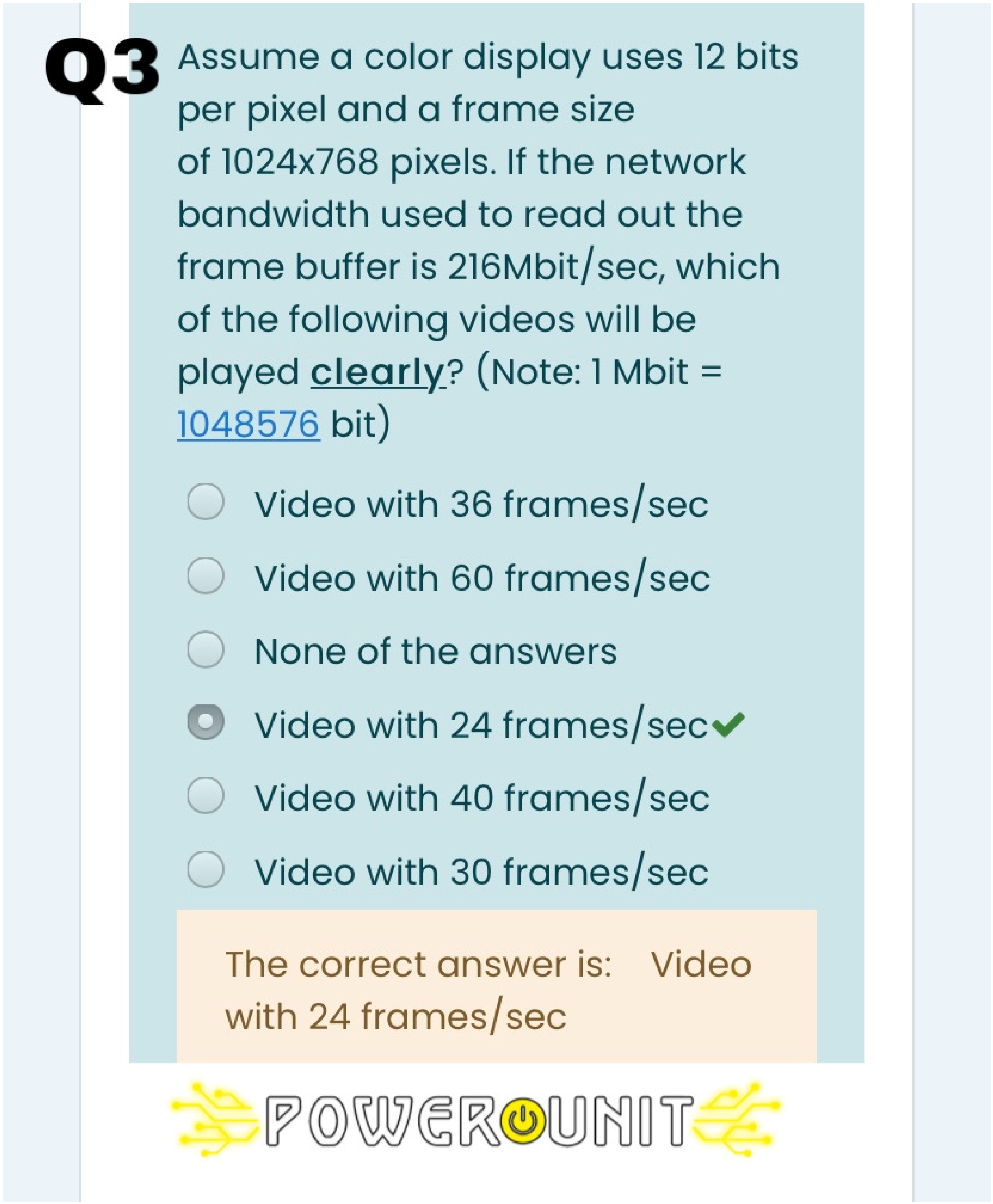


One of the differences between Super Computers and Personal Computers is:

- Personal Computers have smaller fraction of the current computer market
- None of the answers
- Super Computers have stringent performance, power, and reliability requirements
- Personal Computers come with pre-installed software
- [✓ Super Computers are used for programs with high computational and storage demands] ✓

The correct answer is: Super Computers are used for programs with high computational and storage demands

POWERUNIT

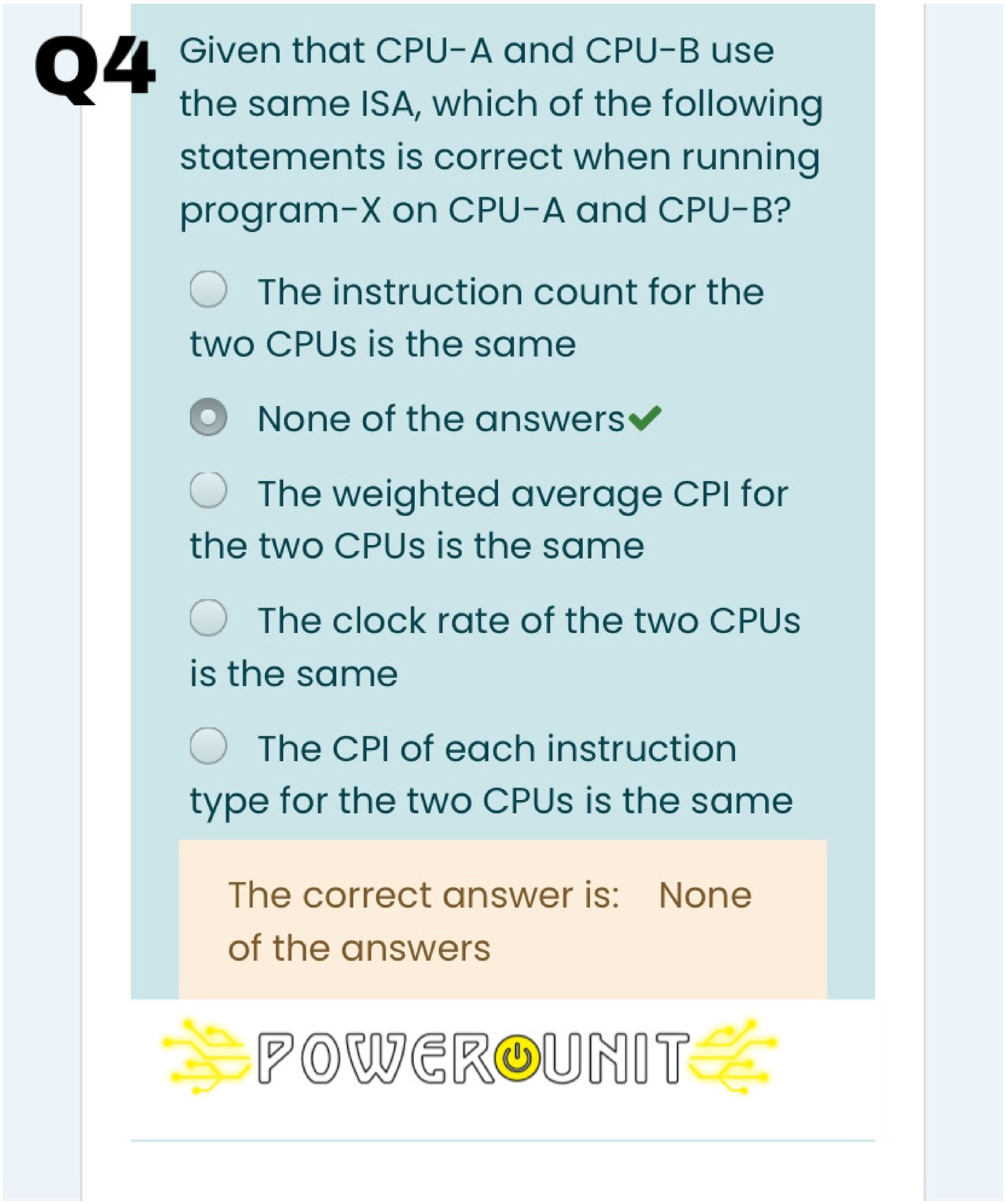


Q3 Assume a color display uses 12 bits per pixel and a frame size of 1024x768 pixels. If the network bandwidth used to read out the frame buffer is 216Mbit/sec, which of the following videos will be played **clearly**? (Note: 1 Mbit = 1048576 bit)

- ○ Video with 36 frames/sec
- ○ Video with 60 frames/sec
- ○ None of the answers
- [✓ Video with 24 frames/sec] ✓
- ○ Video with 40 frames/sec
- ○ Video with 30 frames/sec

The correct answer is: Video with 24 frames/sec

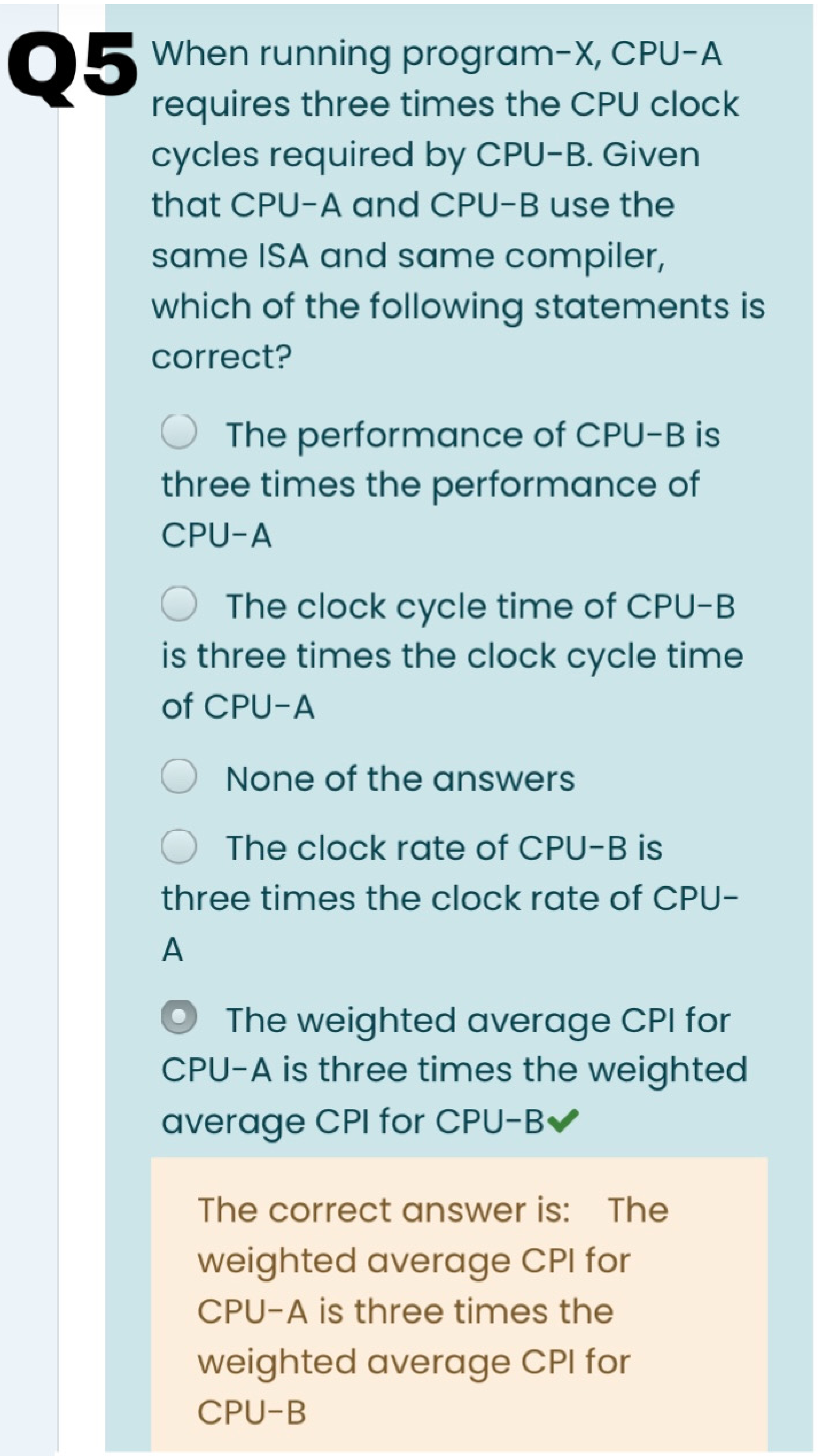
Q4



Given that CPU-A and CPU-B use the same ISA, which of the following statements is correct when running program-X on CPU-A and CPU-B?

- ○ The instruction count for the two CPUs is the same
- [✓ None of the answers] ✓
- ○ The weighted average CPI for the two CPUs is the same
- ○ The clock rate of the two CPUs is the same
- ○ The CPI of each instruction type for the two CPUs is the same

The correct answer is: None of the answers



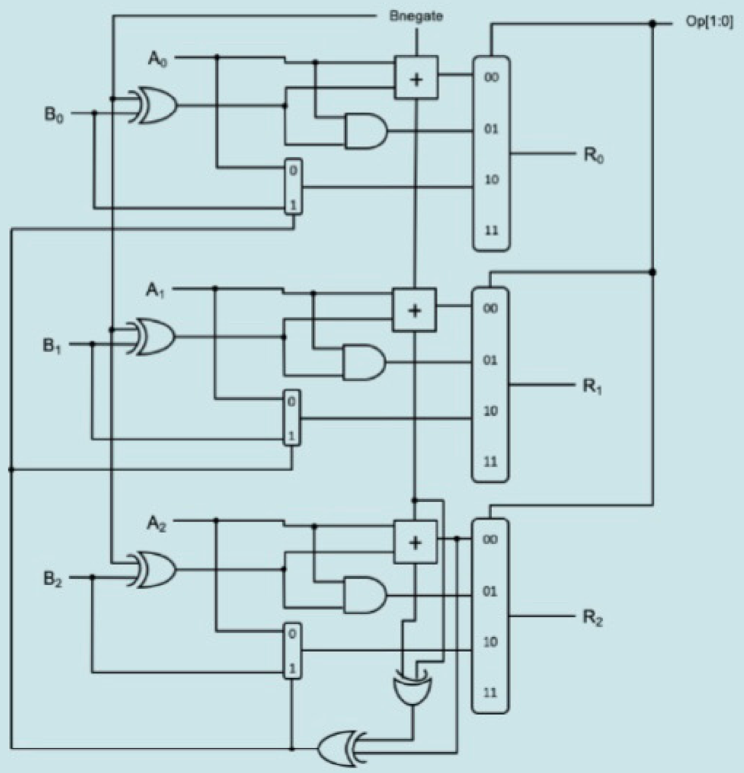
Q5 When running program-X, CPU-A requires three times the CPU clock cycles required by CPU-B. Given that CPU-A and CPU-B use the same ISA and same compiler, which of the following statements is correct?

- ○ The performance of CPU-B is three times the performance of CPU-A
- ○ The clock cycle time of CPU-B is three times the clock cycle time of CPU-A
- ○ None of the answers
- ○ The clock rate of CPU-B is three times the clock rate of CPU-A
- [✓ The weighted average CPI for CPU-A is three times the weighted average CPI for CPU-B] ✓

The correct answer is: The weighted average CPI for CPU-A is three times the weighted average CPI for CPU-B

Q6 A

A 3-bit ALU is designed to perform some arithmetic and logic functions. The ALU is ***partially*** completed below. Answer the following questions accordingly:

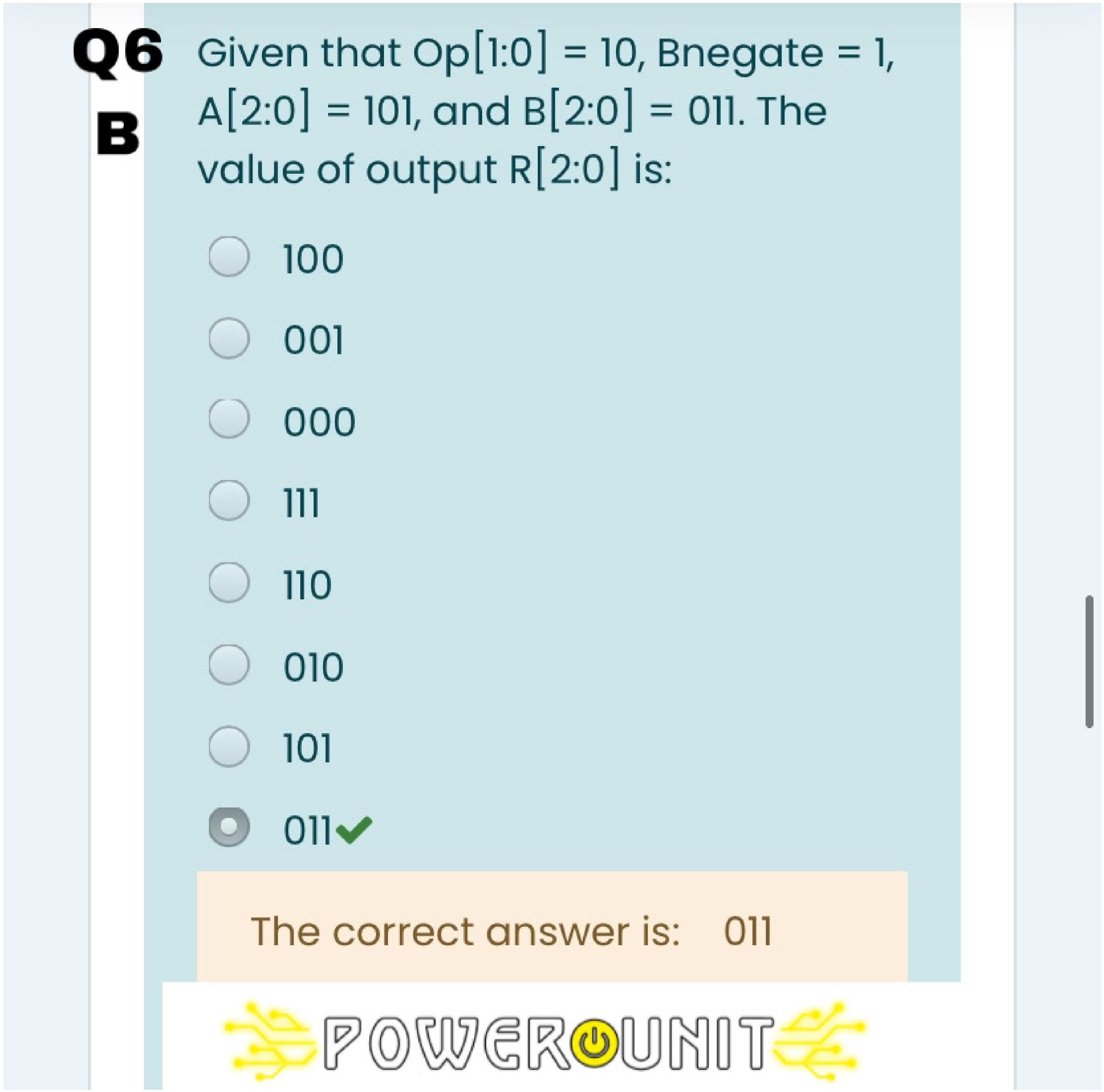


Given that Op[1:0] = 00, Bnegate = 0, A[2:0] = 010, and B[2:0] = 101. The value of output R[2:0] is:

- 001
- [✓ 101] ✗
- 011
- 000
- 010
- 110
- 100
- 111

POWERUNIT

The correct answer is: 111



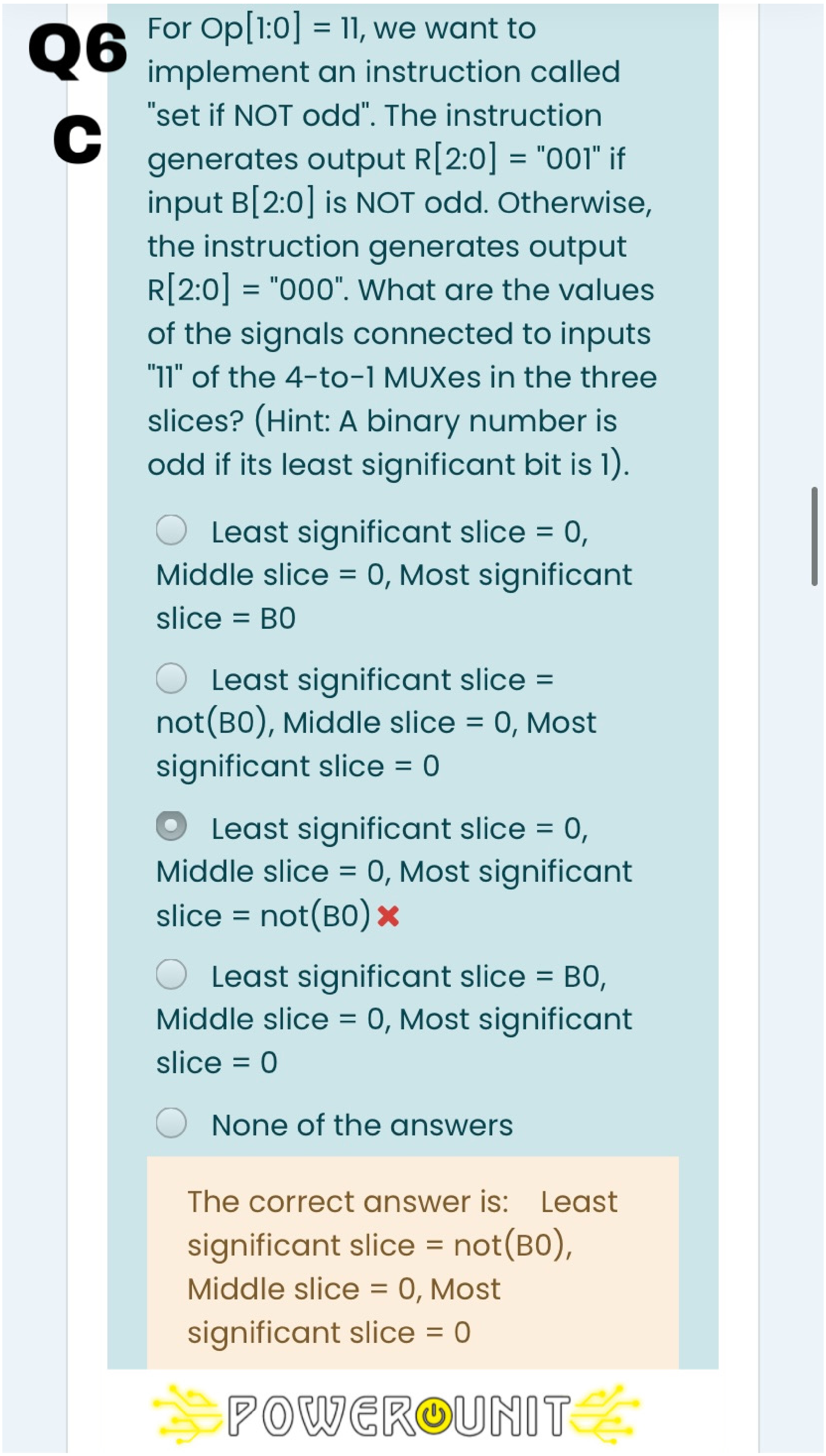
Q6
B

Given that Op[1:0] = 10, Bnegate = 1, A[2:0] = 101, and B[2:0] = 011. The value of output R[2:0] is:

- ○ 100
- ○ 001
- ○ 000
- ○ 111
- ○ 110
- ○ 010
- ○ 101
- [✓ 011] ✓

The correct answer is: 011

POWERUNIT



Q6 c

For Op[1:0] = 11, we want to implement an instruction called "set if NOT odd". The instruction generates output R[2:0] = "001" if input B[2:0] is NOT odd. Otherwise, the instruction generates output R[2:0] = "000". What are the values of the signals connected to inputs "11" of the 4-to-1 MUXes in the three slices? (Hint: A binary number is odd if its least significant bit is 1).

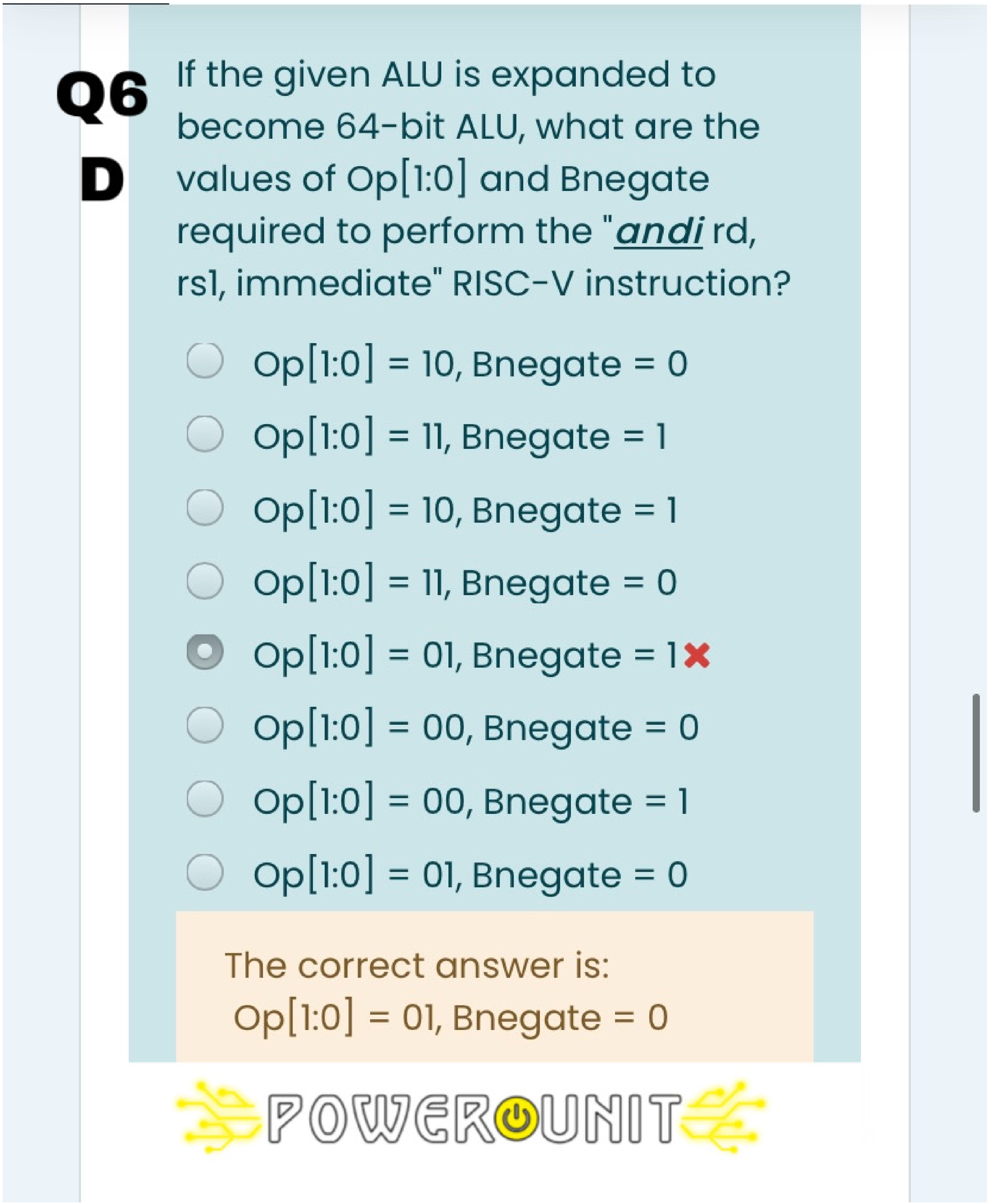
- Least significant slice = 0, Middle slice = 0, Most significant slice = B0
- Least significant slice = not(B0), Middle slice = 0, Most significant slice = 0
- [✓ Least significant slice = 0, Middle slice = 0, Most significant slice = not(B0)] ✗
- Least significant slice = B0, Middle slice = 0, Most significant slice = 0
- None of the answers

The correct answer is: Least significant slice = not(B0), Middle slice = 0, Most significant slice = 0

POWERUNIT

Q6

D

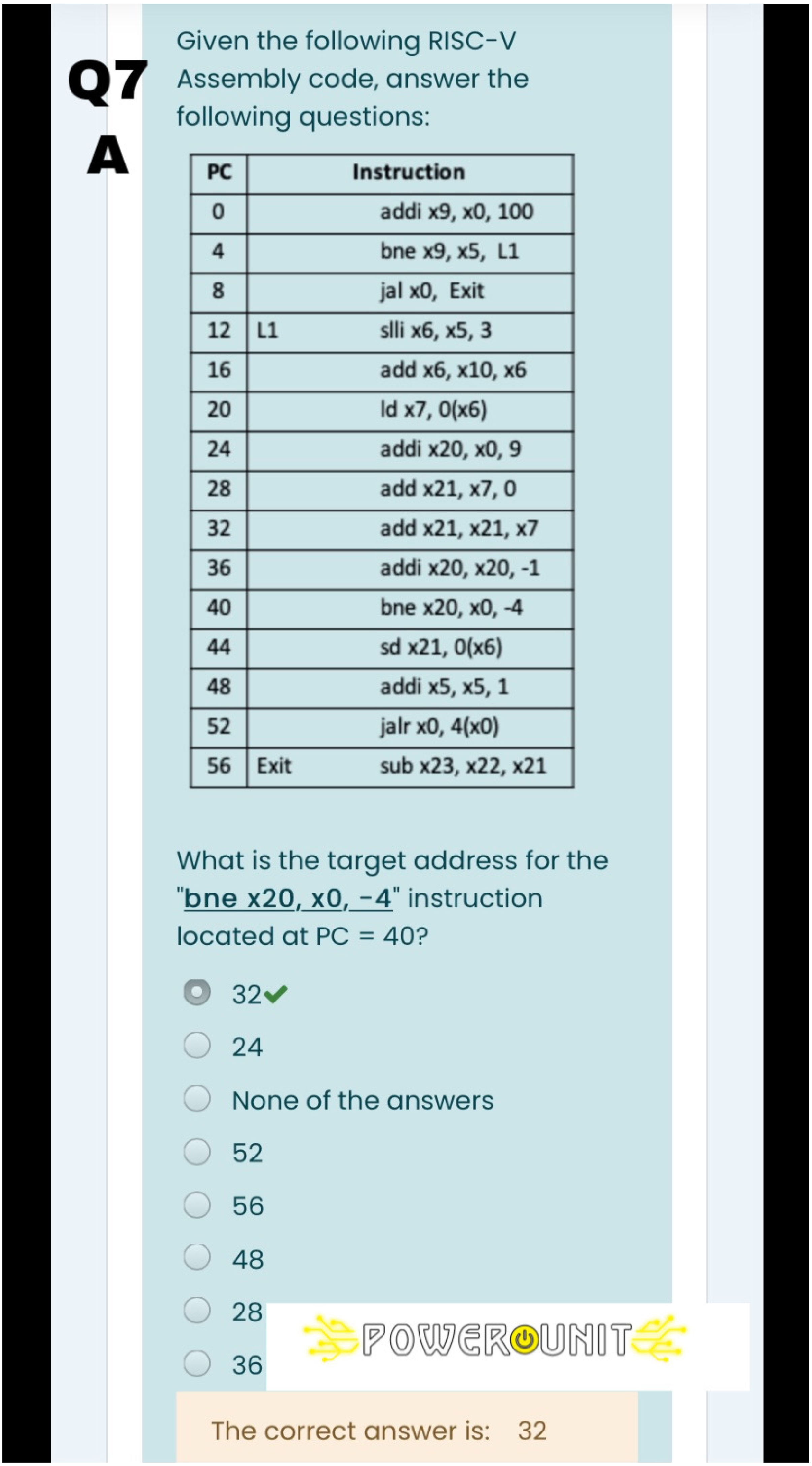


If the given ALU is expanded to become 64-bit ALU, what are the values of Op[1:0] and Bnegate required to perform the "*andi* rd, rs1, immediate" RISC-V instruction?

- Op[1:0] = 10, Bnegate = 0
- Op[1:0] = 11, Bnegate = 1
- Op[1:0] = 10, Bnegate = 1
- Op[1:0] = 11, Bnegate = 0
- [✓ Op[1:0] = 01, Bnegate = 1] ✗
- Op[1:0] = 00, Bnegate = 0
- Op[1:0] = 00, Bnegate = 1
- Op[1:0] = 01, Bnegate = 0

The correct answer is:
Op[1:0] = 01, Bnegate = 0

POWERUNIT



Q7 A

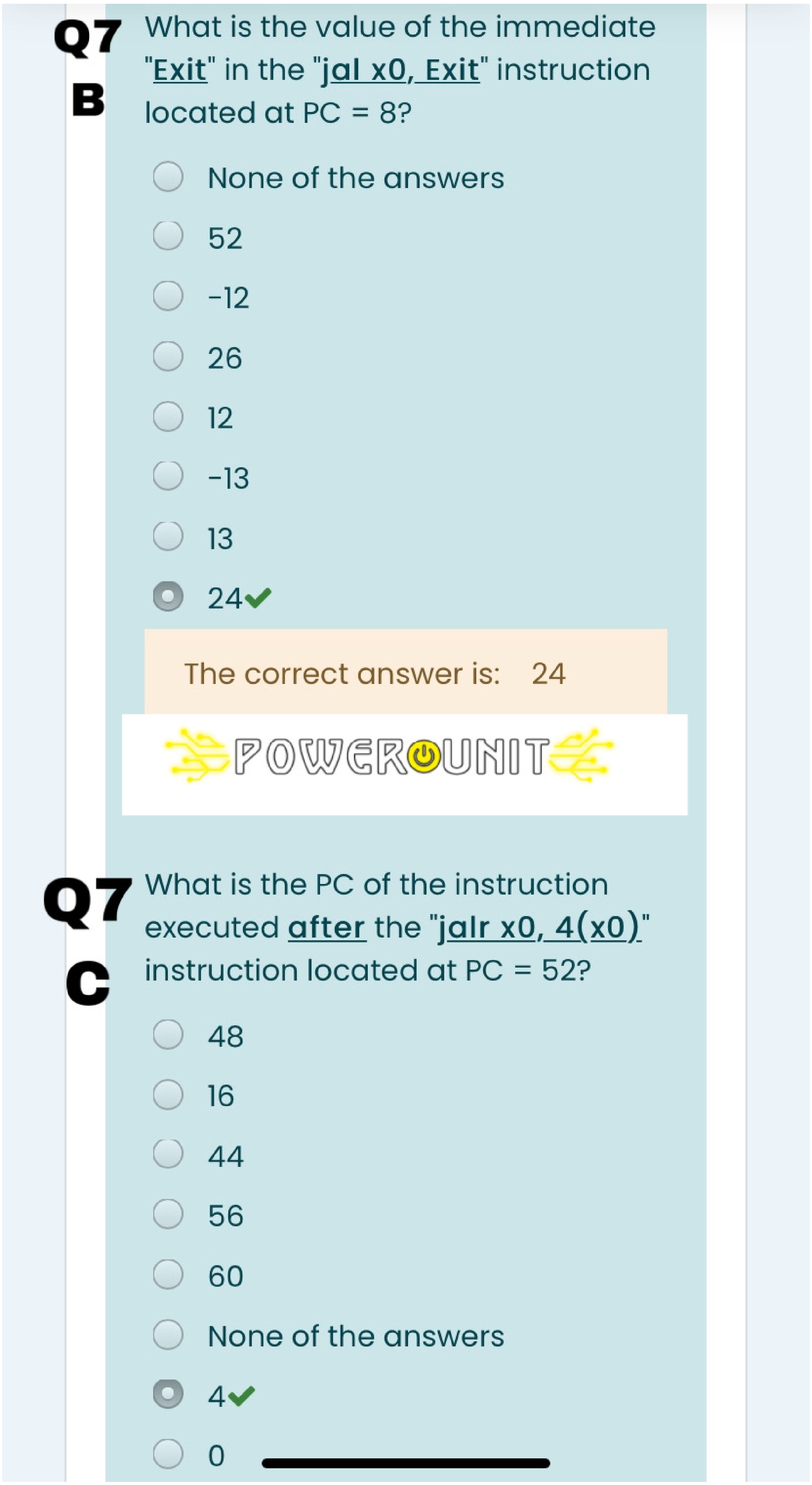
Given the following RISC-V Assembly code, answer the following questions:

PC		Instruction
0		addi x9, x0, 100
4		bne x9, x5, L1
8		jal x0, Exit
12	L1	slli x6, x5, 3
16		add x6, x10, x6
20		ld x7, 0(x6)
24		addi x20, x0, 9
28		add x21, x7, 0
32		add x21, x21, x7
36		addi x20, x20, -1
40		bne x20, x0, -4
44		sd x21, 0(x6)
48		addi x5, x5, 1
52		jalr x0, 4(x0)
56	Exit	sub x23, x22, x21

What is the target address for the "bne x20, x0, -4" instruction located at PC = 40?

- [✓ 32] ✓
- 24
- None of the answers
- 52
- 56
- 48
- 28
- 36

The correct answer is: 32



Q7
B

What is the value of the immediate "Exit" in the "jal x0, Exit" instruction located at PC = 8?

- None of the answers
- 52
- -12
- 26
- 12
- -13
- 13
- [✓ 24] ✓

The correct answer is: 24

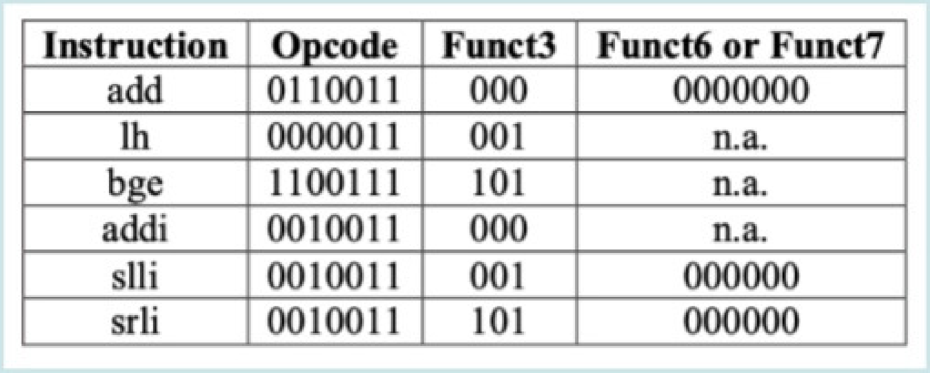
POWERUNIT

Q7
C

What is the PC of the instruction executed **after** the "jalr x0, 4(x0)" instruction located at PC = 52?

- 48
- 16
- 44
- 56
- 60
- None of the answers
- [✓ 4] ✓
- 0

Q8 Given the following operational codes and function fields:



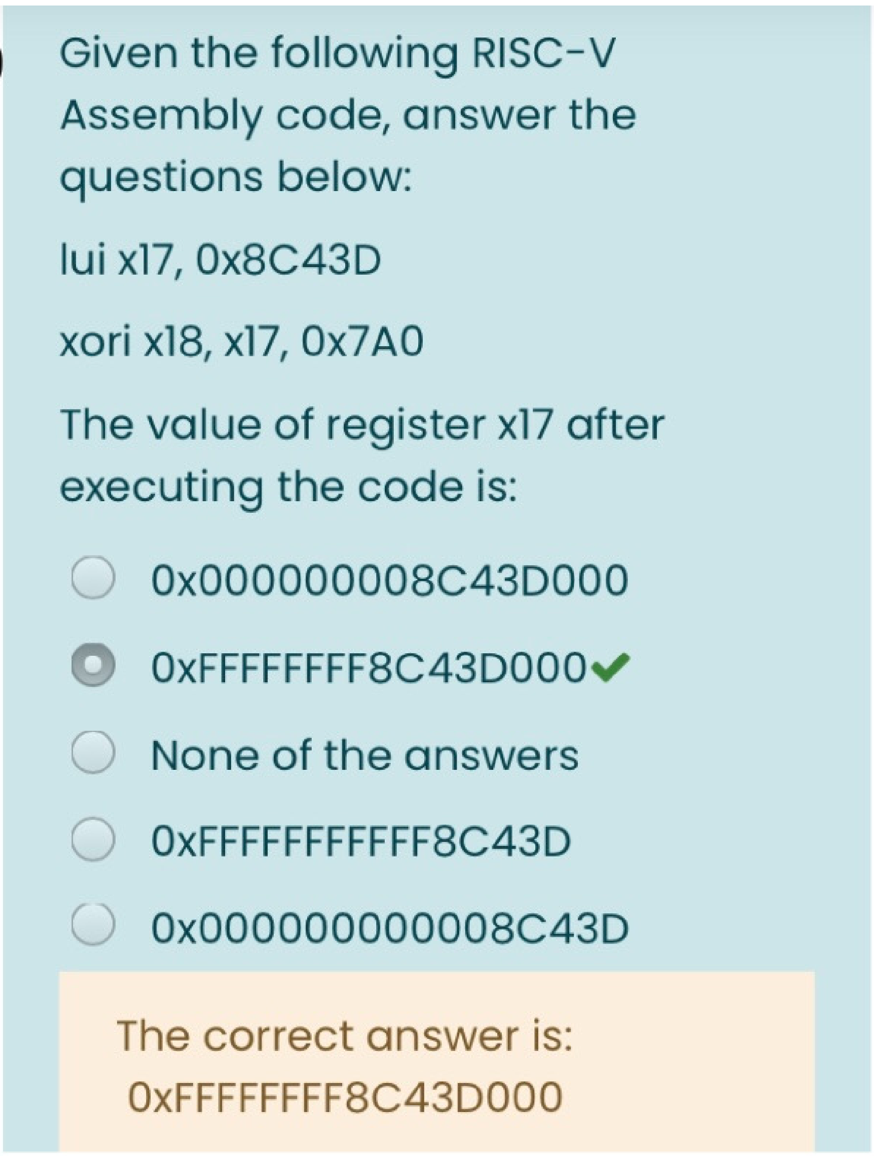
Instruction	Opcode	Funct3	Funct6 or Funct7
add	0110011	000	0000000
lh	0000011	001	n.a.
bge	1100111	101	n.a.
addi	0010011	000	n.a.
slli	0010011	001	000000
srli	0010011	101	000000

What is the machine code of the instruction "slli x5, x7, 4" ?

- [✓ 0x00439293] ✓
- 0x00429393
- 0x00429C93
- None of the answers
- 0x00439493

The correct answer is:
0x00439293

Q9 A



Given the following RISC-V Assembly code, answer the questions below:

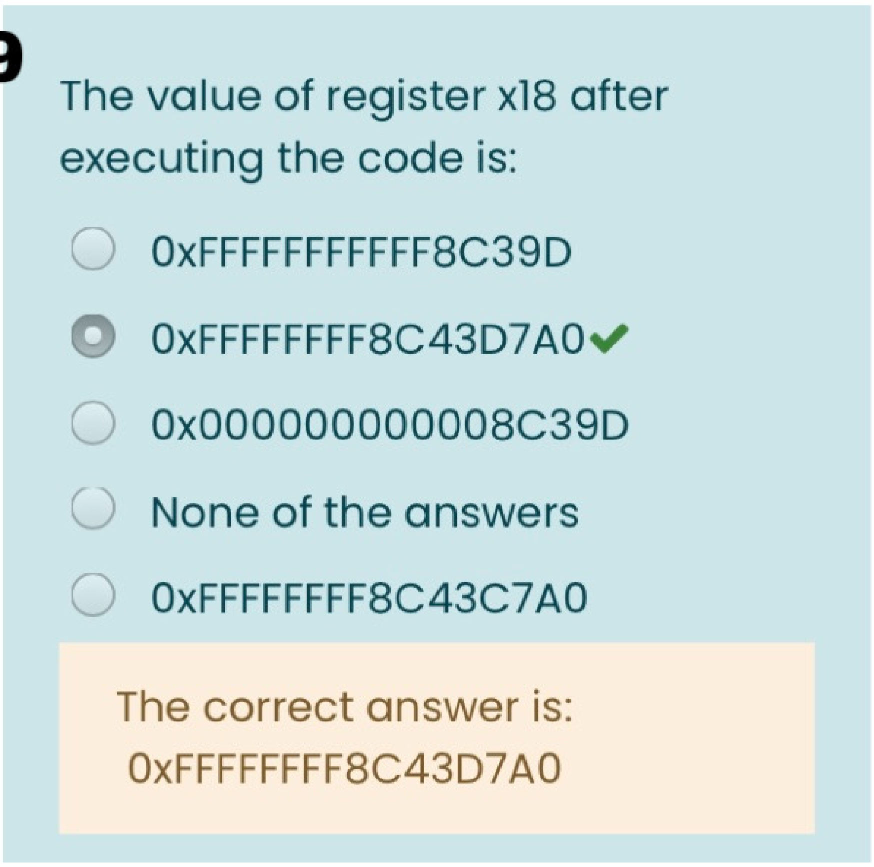
```
lui x17, 0x8C43D

xori x18, x17, 0x7A0
```

The value of register x17 after executing the code is:

- 0x000000008C43D000
- [✓ OxFFFFFFFF8C43D000] ✓
- None of the answers
- OxFFFFFFFFFFF8C43D
- 0x000000000008C43D

The correct answer is:
OxFFFFFFFF8C43D000

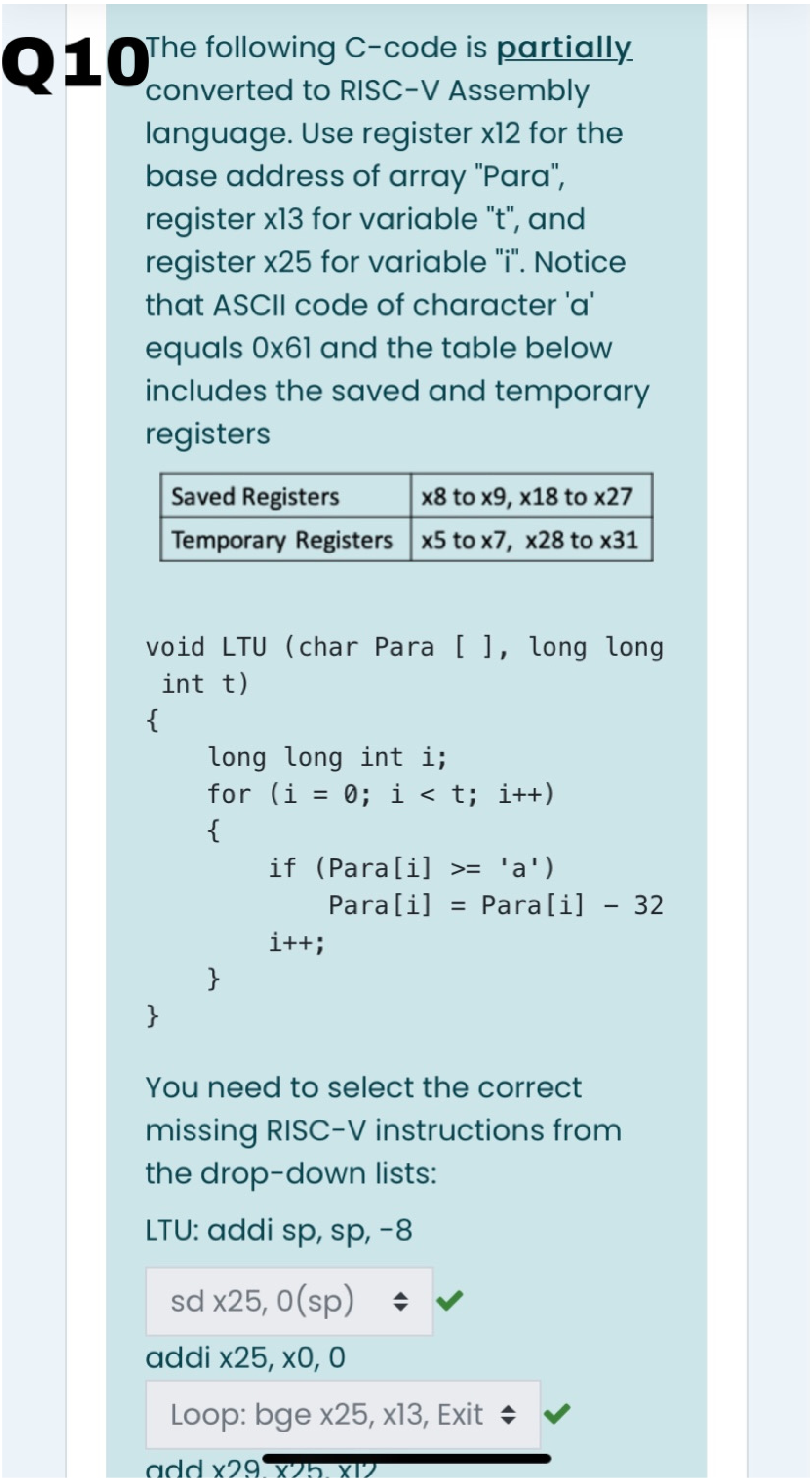
Q9 B



The value of register x18 after executing the code is:

- OxFFFFFFFFFFF8C39D
- [✓ OxFFFFFFFF8C43D7A0] ✓
- 0x000000000008C39D
- None of the answers
- OxFFFFFFFF8C43C7A0

The correct answer is:
OxFFFFFFFF8C43D7A0



Q10 The following C-code is partially converted to RISC-V Assembly language. Use register x12 for the base address of array "Para", register x13 for variable "t", and register x25 for variable "i". Notice that ASCII code of character 'a' equals 0x61 and the table below includes the saved and temporary registers

Saved Registers	x8 to x9, x18 to x27
Temporary Registers	x5 to x7, x28 to x31

```
void LTU (char Para [ ], long long
 int t)
{
    long long int i;
    for (i = 0; i < t; i++)
    {
        if (Para[i] >= 'a')
            Para[i] = Para[i] - 32
        i++;
    }
}
```

You need to select the correct missing RISC-V instructions from the drop-down lists:

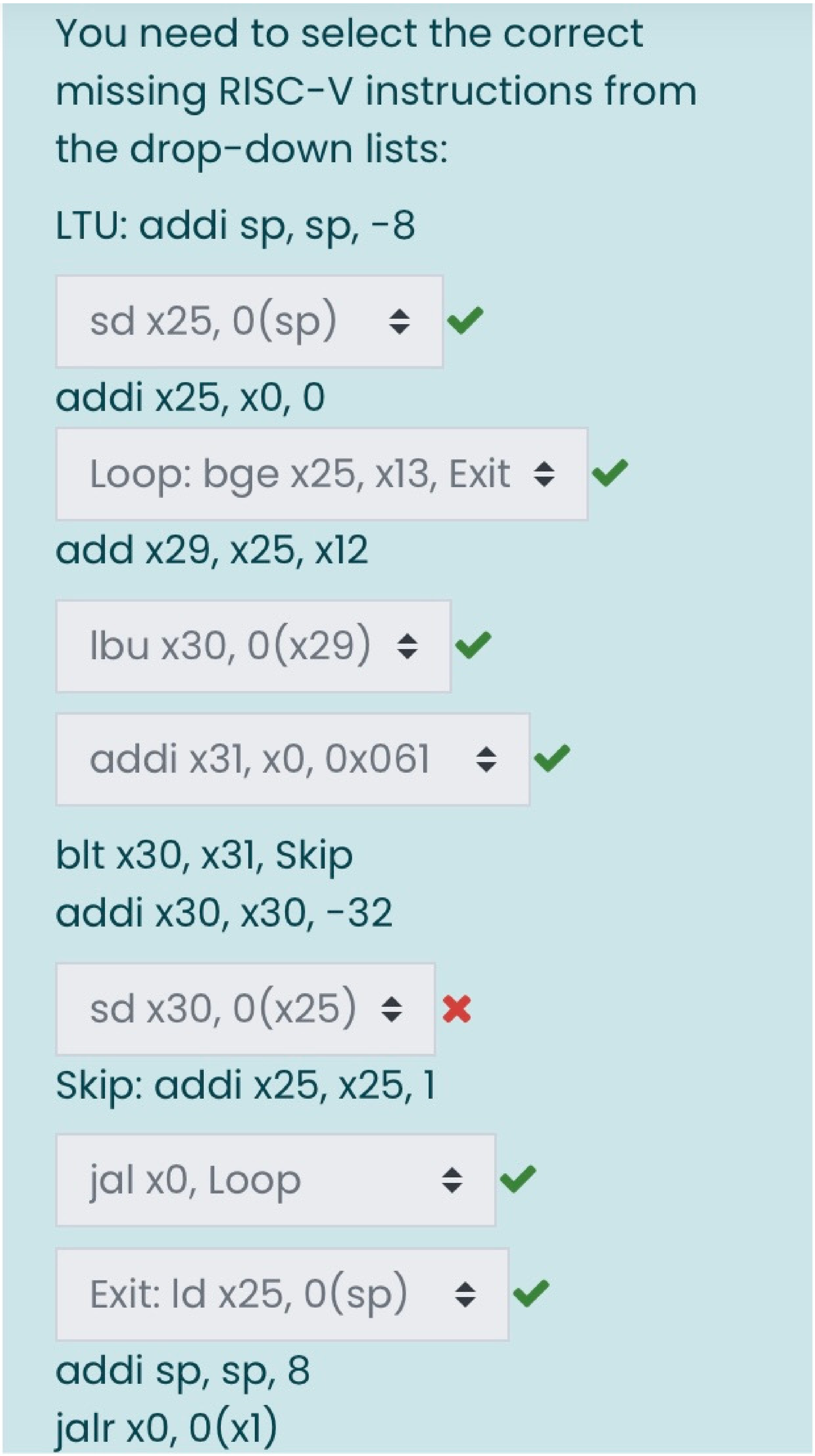
LTU: addi sp, sp, -8

sd x25, 0(sp) ✓

addi x25, x0, 0

Loop: bge x25, x13, Exit ✓

add x29, x25, x12



You need to select the correct missing RISC-V instructions from the drop-down lists:

```
LTU: addi sp, sp, -8
[sd x25, 0(sp)] ✓
addi x25, x0, 0
[Loop: bge x25, x13, Exit] ✓
add x29, x25, x12
[lbu x30, 0(x29)] ✓
[addi x31, x0, 0x061] ✓
blt x30, x31, Skip
addi x30, x30, -32
[sd x30, 0(x25)] ✗
Skip: addi x25, x25, 1
[jal x0, Loop] ✓
[Exit: ld x25, 0(sp)] ✓
addi sp, sp, 8
jalr x0, 0(x1)
```

Q11

The following tables include: Non-leaf procedure "Srch" written in RISC-V assembly language, a procedure call, and memory contents. The "Srch" procedure has four arguments mapped to registers x11, x12, x13 and x14. The return value of the procedure is mapped to register x10. Accordingly, answer the questions below:

Procedure "Srch"		
PC	**Instruction**	
40	Srch:	addi sp, sp, -8
44		sd x1, 0(sp)
48		bge x13, x12, L1
52		addi x10, x0, -1
56		beq x0, x0, Exit
60	L1:	slli x5, x12, 3
64		add x5, x5, x11
68		ld x5, 0(x5)
72		bne x5, x14, L2
76		add x10, x12, x0
80		beq x0, x0, Exit
84	L2:	slli x6, x13, 3
88		add x6, x6, x11
92		ld x6, 0(x6)
96		bne x6, x14, L3
100		add x10, x13, x0
104		beq x0, x0, Exit
108	L3:	addi x12, x12, 1
112		addi x13, x13, -1
116		jal x1, Srch
120	Exit:	ld x1, 0(sp)
124		addi sp, sp, 8
128		jalr x0, 0(x1)

Procedure Call	
PC	**Instruction**
0	addi x11, x0, 0
4	addi x12, x0, 0
8	addi x13, x0, 9
12	addi x14, x0, 0x732
16	jal x1, Srch

Address	Memory Contents
72 to 79	0x00000000000005F3
64 to 71	0xFFFFFFFFFFFFF900
56 to 63	0x00000000000001EB
48 to 55	0xFFFFFFFFFFFFFC79
40 to 47	0x00000000000004AC
32 to 39	0xFFFFFFFFFFFFFA15
24 to 31	0xFFFFFFFFFFFFF86D
16 to 23	0x0000000000000732
8 to 15	0x0000000000000025
0 to 7	0x00000000000003B1

124	addi sp, sp, 8
128	jalr x0, 0(x1)

Q11 A The first value pushed in the stack is:

120 16 [✓ 20] ✓ 116 0

The correct answer is: 20

Q11 B How many values are pushed in the stack?

3 5 2 4 6 [✓ 1] ✗

The correct answer is: 3

Q11 C The return value from the given procedure call is:

4 8 16 5 9 7 2 -1 56 1 72 0

The correct answer is: 2

POWERUNIT