Review Test Submission: Final Exam

User	Mualla Argin
Course	21 SPRING CSCE 312 200,500-503,506-513: CSCE 312 SP 2021
Test	Final Exam
Started	5/3/21 2:00 PM
Submitted	5/3/21 4:30 PM LATE
Due Date	5/3/21 4:00 PM
Status	Completed
Attempt Score	46 out of 90 points
Time Elapsed	2 hours, 30 minutes

Instructions

Regarding the Questions

Please read each question carefully before answering. We will be unable to accommodate answers that do not follow the expected format.

Guide for Questions to ease understanding of emphasis:

- Main Task is written in **RED Boldface**
- Key points are written in **Black Boldface**
- Advisory notes are prefixed with "Note" and written in *Blue Italics.*

Please be sure to read the advisory notes for "fill-in-the-blank" questions very carefully. These questions are auto-graded so it is important to adhere to the stated advisory for entering answers.

Regarding the Timing

A 15 minute buffer will be added to the finish time to accommodate any administrative overhead related to e-campus submission. Use this time for submission contingencies only! Submissions received on e-campus after the buffer will incur 5% penalty for every minute late.

Advisory: ***Please do NOT wait to submit the exam at the last minute as it places you at jeopardy of missing the deadline for online submission.***

Results Displayed	All Answers, Submitted Answers, Correct Answers, Feedback, Incorrectly Answered Questions

Question 1

1 out of 1 points

On my honor, as an Aggie, I promise to abide by the Academic Code of Conduct specified in the course syllabus and will follow the honor code policies stated in the exam preparation guide.

Selected Answer: ✓ True

Answers: ✓ True

False

Question 2

1 out of 1 points

Which of the following sequence(s) of Hack assembly statements result(s) in RAM[17] = 2 ? Mark **ALL** correct answers.

***Note**: Multiple HACK instructions in each sequence are separated by a comma.*

Selected Answers: @17, M=1, M=M+1

✓

✓ @1, D=A+1, @17, M=D

@2, D=A, @17, M=D

✓

Answers: @17, M=2

@17, M=1, M=M+1

✓

✓ @1, D=A+1, @17, M=D

@2, D=A, @17, M=D

✓

Question 3

1 out of 1 points

Which one of the given choices is **FALSE** after running the following two-line HACK program?

```
@1
M=A-1; JEQ
```

Selected Answer: ✓ B. Register R0 (i.e. RAM[0]) has the value 1

Answers:

A. The next instruction to be executed is instruction at line 1 in the ROM

✓ B. Register R0 (i.e. RAM[0]) has the value 1

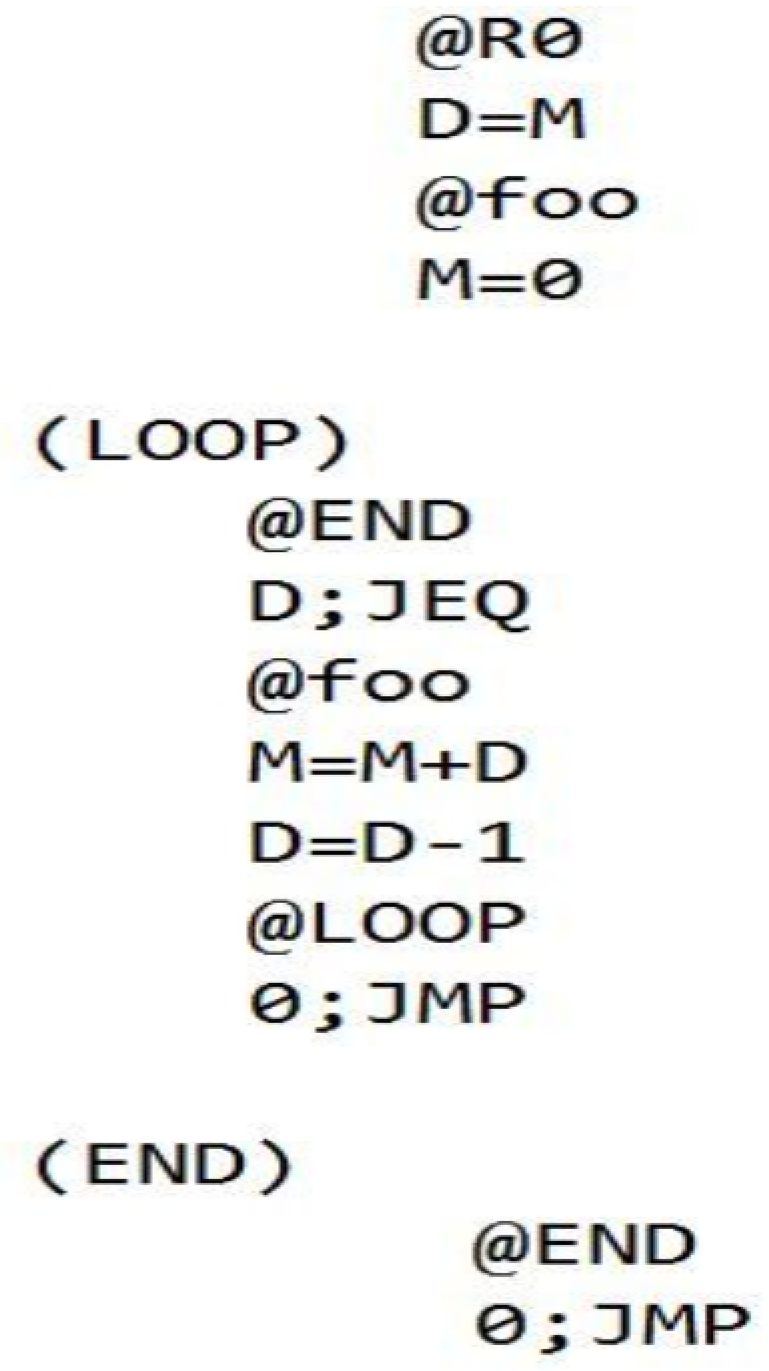
C. Register R1 (i.e. RAM[1]) has the value 0

D. Register A has value 1

Question 4

2 out of 2 points

What would be the value in RAM location assigned to variable **foo** if **R0 (i.e. RAM[0]) contained the value 10** and the following HACK program is executed?



```
                @R0
                D=M
                @foo
                M=0

(LOOP)
         @END
         D;JEQ
         @foo
         M=M+D
         D=D-1
         @LOOP
         0;JMP

(END)
                    @END
                    0;JMP
```

Selected Answer: ✓ C. 55

Answers:

A. 15

B. 10

✓ C. 55

D. 50

Question 5

2 out of 6 points

A palindrome is a sequence of characters which reads the same forward and backward, such as "madam" or "racecar" or "12321".

The following HACK ASM program ***claims*** to check for the occurrence of a palindrome in a sequence of 5 numbers stored in 5 RAM locations represented by variables **a b c d e** in that order. The program ***claims*** to write the number 1 in RAM location f if a palindrome is detected, and the number -1 in RAM location f otherwise.

Note: For example, if RAM locations a b c d e contain 5 7 3 7 5 respectively, then the program must end with the number 1 in location f. On the other hand, if RAM locations a b c d e contain 1 0 1 1 0 respectively, then the program must end with the number -1 in location f.

Unfortunately, the program has one or more bugs. **Your task is to identify the bug(s) and state the correction(s).** *Each instruction is annotated with a line number for your convenience.*

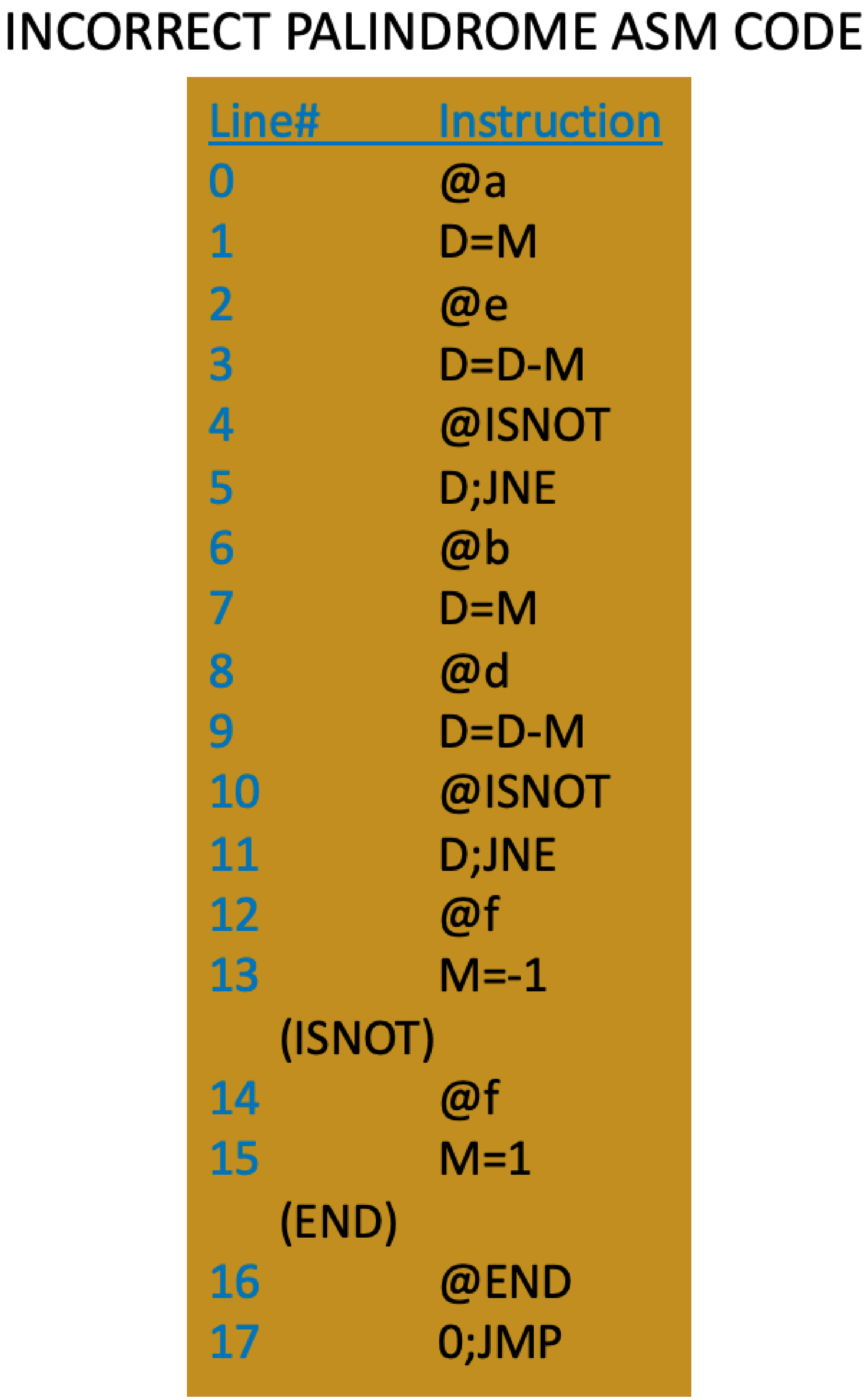
Write your answer along with a summary explaining your reasoning in the space provided below.

Note: Some examples of how you may write your answer are shown below for hypothetical scenarios:

- ***Eg.1****: If there is a mistake in instruction at Line 3, then you could write as "Line3: D=D-A".*
- **Eg.2:** *If the mistake is in missing instructions, then you could write as "Missing instructions: Insert 2 instructions D=M and M=D after Line 11"*

Note: Your answers will be manually graded so the above suggestions are just for ease in communication.

Full points will only be awarded for correct answers with minimum modification to the code.



INCORRECT PALINDROME ASM CODE

```
Line#        Instruction
0            @a
1            D=M
2            @e
3            D=D-M
4            @ISNOT
5            D;JNE
6            @b
7            D=M
8            @d
9            D=D-M
10           @ISNOT
11           D;JNE
12           @f
13           M=-1
     (ISNOT)
14           @f
15           M=1
     (END)
16           @END
17           0;JMP
```

Selected Answer:	Missing Instructions : Add JMP statement after Line 13 so program knows when to M=-1
Correct Answer:	✓

(ANSWER) Corrections are as follows:
#1: Line 13: M=1
#2: Line 15: M=-1
#3: Insert the following two lines after Line 13 so as to prevent (ISNOT) routine from executing when a palindrome is detected

```
        @END
        0;JMP
```

Response Feedback: [None Given]

Question 6

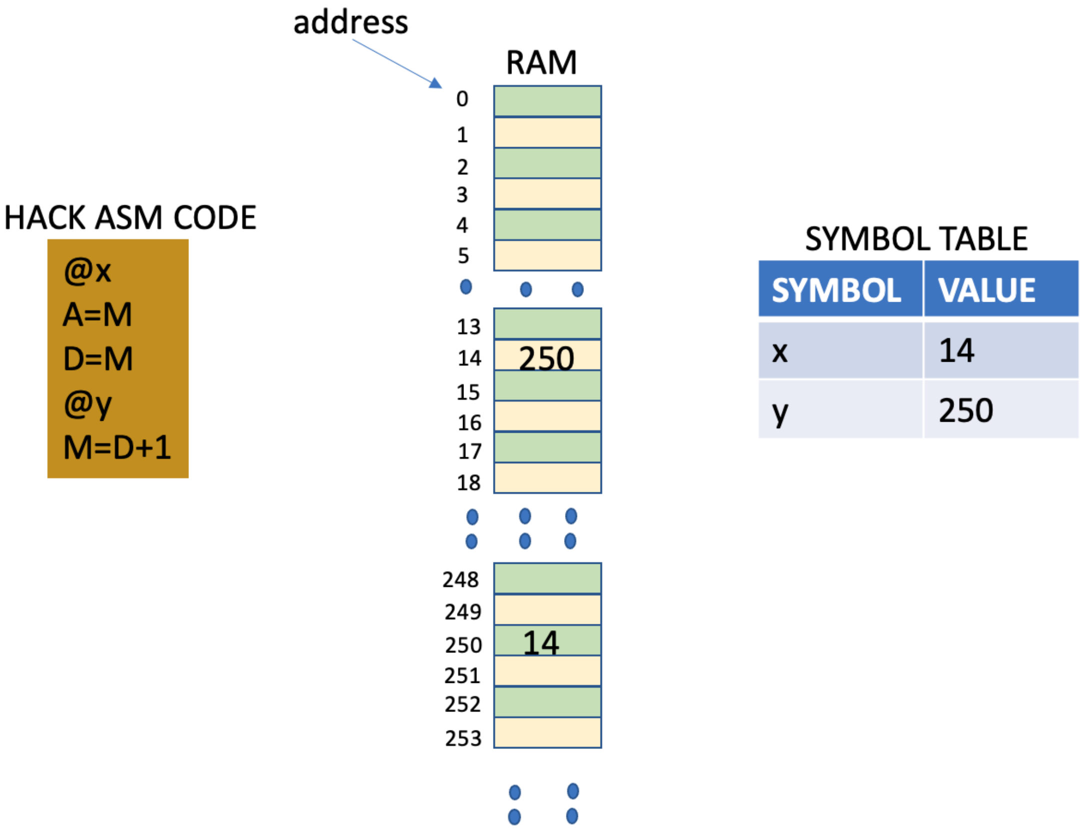
6 out of 6 points

What will be the content of RAM[250] at the end of execution of the HACK ASM code shown below?

Write your answer along with a short explanation in the space provided. Correct answer without explanation will only receive partial credits.

Note:

- *Only the relevant entries of the symbol table are shown.*
- *A snapshot of the RAM are shown along with each RAM register address. The dots between RAM registers are simply there to demonstrate the expanse of the RAM without showing every register location.*



HACK ASM CODE

```
@x
A=M
D=M
@y
M=D+1
```

SYMBOL TABLE

SYMBOL	VALUE
x	14
y	250

Selected Answer: RAM[250] = 15

Correct Answer: ✓

RAM[250] will be 15. The following outlines the flow of the program with the use of comments.

```
@x // A = 14 from the symbol table
A = M //A = M[x] --> A = M[14] --> A = 250
D = M // D = M[250] --> D = 14
@y //A = 250 from the symbol table
M = D + 1 //M[y] = 14 + 1 --> M[250] = 15
```

Response Feedback: [None Given]

Question 7

4 out of 5 points

Write HACK assembly code for the following high level function:

`temp = RAM[RAM[i]+1]`

Note:

- Please write one HACK instruction per line.
- Feel free to add comments to your code.

Full marks will only be awarded for correct answers with fewest lines of ASM code.

Selected Answer:

```
@ i
A = M + 1
D = M
@temp
M = D
```

Correct Answer:

`temp = RAM[RAM[i]+1]`

```
@i  // A = address of variable i from the symbol table
A = M // A = variable i's memory value
A = M + 1 //A = RAM[i]+1
D = M //D = RAM[RAM[i]+1
@temp // A = address of variable i from the symbol table
M = D
```

✓

Response Feedback: [None Given]

Question 8

0 out of 3 points

✗ The output of Program Counter (PC) in a computer's CPU is sent as an address to the computer’s (instruction) ROM to fetch instructions. If each register in the ROM is **32 bits** and the ROM is **64 KB** (Kilo Byte), **the number of address bits to the ROM to fetch an instruction from the selected register is**

Selected Answer: ✗ B. 5

Answers: ✓ A. 14

B. 5

C. 16

D. 10

Response Feedback:

Given: Each register in the ROM is **32 bits** and the ROM is **64 KB** (Kilo Byte).

The number of address bits to the ROM to fetch an instruction from the selected register can be calculated as follows:

Number of ROM entries = 64 * 2^10 * 8 bits / 32 bits = 16 * 2^10 entries

Number of address bits = log (16 * 2^10) = log (2^4 * 2^10) = log (2^14) = 14

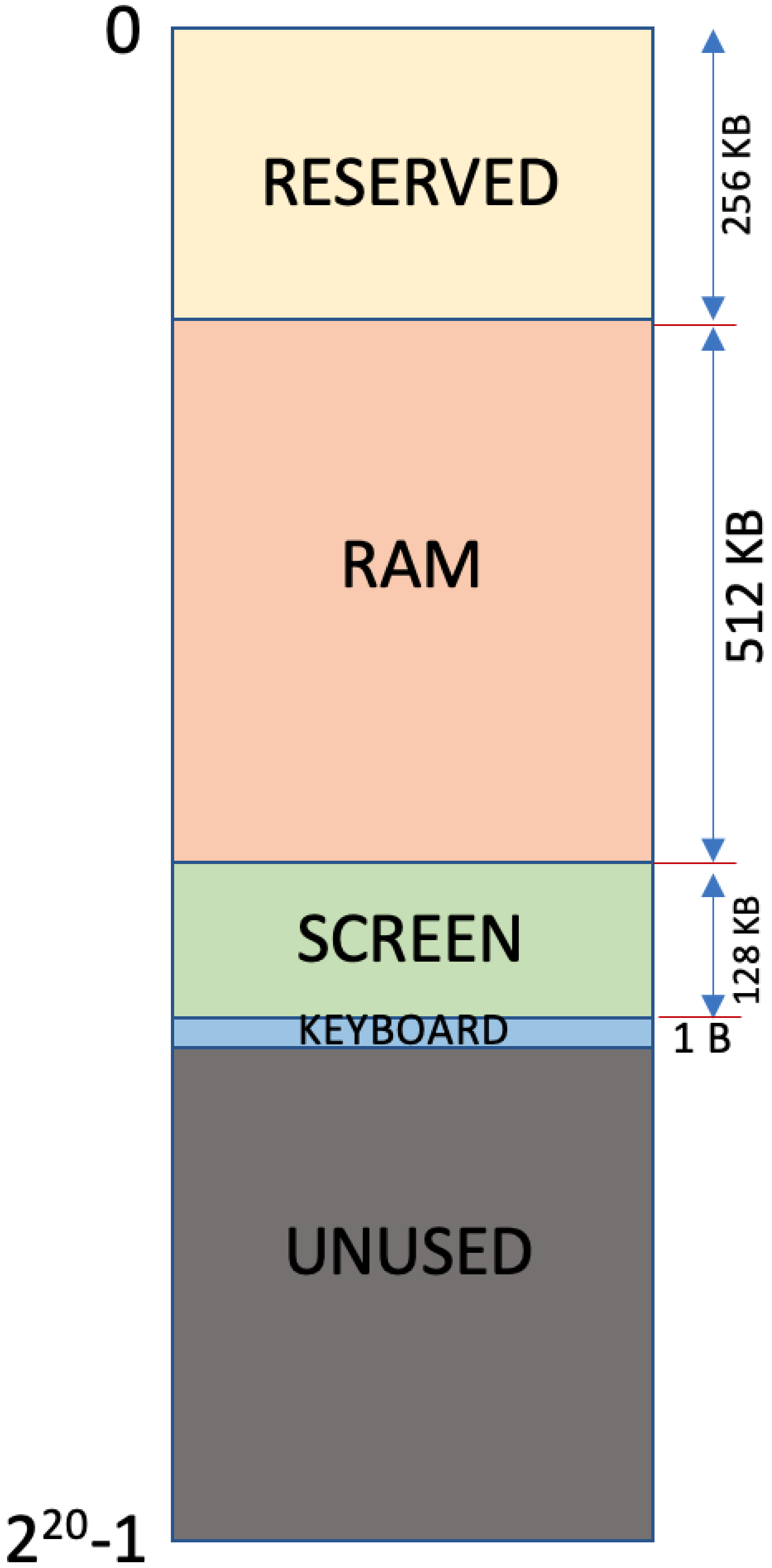
Question 9

0 out of 8 points

The regions of a **1 MB** (Mega Byte) Memory are shown in the diagram below.

Each memory register is **1B** (Byte).

The memory regions are as follows: **256 KB RESERVED, 512KB RAM, 128KB SCREEN, and 1B KEYBOARD.** Their arrangement is shown in the diagram below.


0
RESERVED
256 KB
RAM
512 KB
SCREEN
128 KB
KEYBOARD
1 B
UNUSED
2^20-1

Fill in the blanks to submit your answer to the following questions:

(a) **The lowest memory address associated with the 512KB RAM is [A]. Write the complete 20-bit address in HEXADECIMAL notation without spaces.** *Note: For example, the lowest memory address associated with the RESERVED region is 0x00000 in HEXADECIMAL notation.*

(b) **The highest memory address associated with the 512KB RAM is [B]. Write the complete 20-bit address in HEXADECIMAL notation without spaces.**

(c) **The memory-mapped IO address of the 1B Keyboard register is [C]. Write the complete 20-bit address in HEXADECIMAL notation without spaces.**

(d) **Say we utilize a DMUX8Way to route the load signal to the various regions of the memory in order to perform a data write. What would be the binary values of the DMUX8Way select bits sel[2:0] needed in order to perform data write to SCREEN? [D].** *Note: For example, if the select bits 2:0 were 011 then you'd write 0b011 or 011*

Specified Answer for: A	✗	40000B4444
Specified Answer for: B	✗	0x27FFF
Specified Answer for: C	✗	0x27FF0
Specified Answer for: D	✗	001

Correct Answers for: A		
Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	40000	
✓ *Exact Match*	0x40000	
Correct Answers for: B		
Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	BFFFF	
✓ *Exact Match*	0xBFFFF	
Correct Answers for: C		
Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	E0000	
✓ *Exact Match*	0xE0000	
Correct Answers for: D		
Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	110	
✓ *Exact Match*	0b110	

Response Feedback: Range of the 20-bit Addresses is as follows:

	Start	End
Reserved.	0x00000	0x3FFFF
RAM	0x40000	0xBFFFF
SCREEN	0xC0000	0xDFFFF

KEYBOARD	0xE0000	N/A

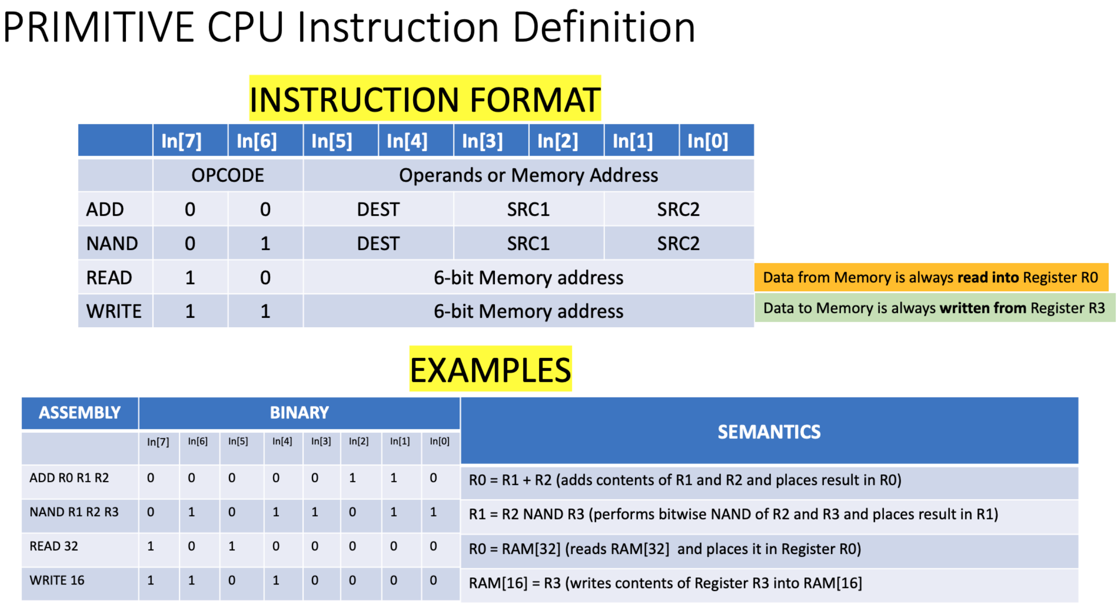
The select bits for DMUX8Way to do SCREEN selection is 0b110 based on the above address range

Question 10

0 out of 14 points

The diagrams below shows the instruction definition and schematic of a primitive CPU.

Study the instruction format carefully (along with the example illustrations). Then study the schematic diagram closely and convince yourself of its correct interpretation of the instruction definition.



PRIMITIVE CPU Instruction Definition

INSTRUCTION FORMAT

	In[7]	In[6]	In[5]	In[4]	In[3]	In[2]	In[1]	In[0]
	OPCODE		Operands or Memory Address					
ADD	0	0	DEST		SRC1		SRC2	
NAND	0	1	DEST		SRC1		SRC2	
READ	1	0	6-bit Memory address					
WRITE	1	1	6-bit Memory address					

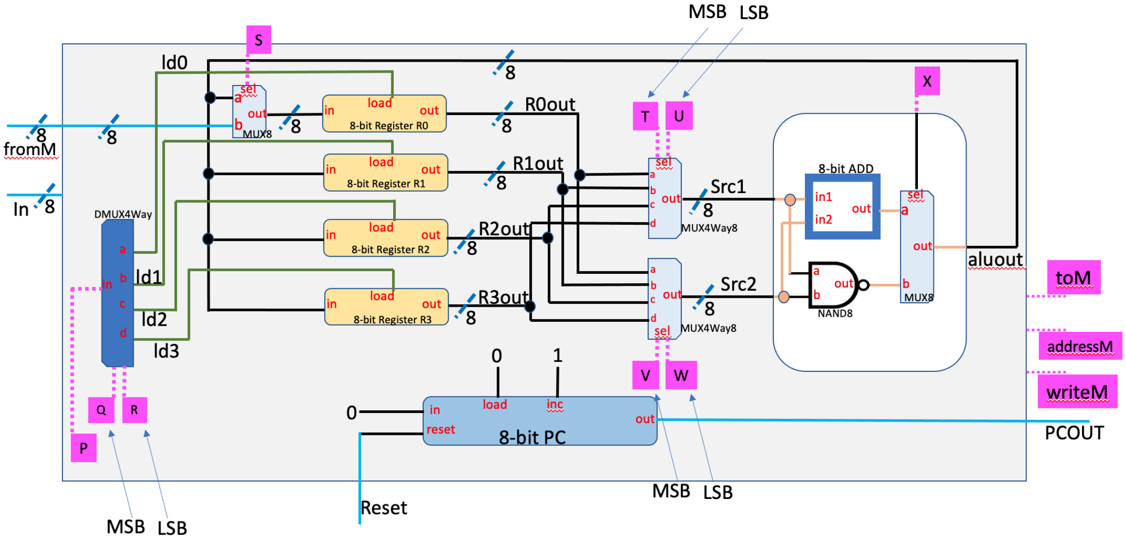
Data from Memory is always **read into** Register R0

Data to Memory is always **written from** Register R3

EXAMPLES

ASSEMBLY	BINARY								SEMANTICS
	In[7]	In[6]	In[5]	In[4]	In[3]	In[2]	In[1]	In[0]	
ADD R0 R1 R2	0	0	0	0	0	1	1	0	R0 = R1 + R2 (adds contents of R1 and R2 and places result in R0)
NAND R1 R2 R3	0	1	0	1	1	0	1	1	R1 = R2 NAND R3 (performs bitwise NAND of R2 and R3 and places result in R1)
READ 32	1	0	1	0	0	0	0	0	R0 = RAM[32] (reads RAM[32] and places it in Register R0)
WRITE 16	1	1	0	1	0	0	0	0	RAM[16] = R3 (writes contents of Register R3 into RAM[16])

Primitive CPU Schematic Diagram



READ THE FOLLOWING RULES VERY CAREFULLY BEFORE WRITING YOUR ANSWERS:

Your task is to fill in the blanks for the logic corresponding to all magenta boxes that are also marked with dotted lines in the schematic diagram. The content for fill in the blank may be a logic equation or a single or multi-bit signal.

RULES:

1. Datapath is 8-bit wide. 0 is the Least Significant Bit (LSB) index and 7 is the Most Significant Bit (MSB) index. **Individual signals of a bus are represented without parentheses (example: In7 for In[7]). Buses are represented as Bus[High:Low] (example: In[7:0]).**

2. **You may ONLY use AND, OR, NOT operations to write logic expressions.** No other operations are allowed.

3. Leave a **single space** on both sides of AND/OR binary operators e.g. x **AND** y, x **OR** y

4. Use **~ symbol** for **NOT** of a signal. Put ~ around parentheses, e.g. **~(z)** for NOT operation on z, ~(In6) for NOT operation on In6.

5. Use 1 or 0 to denote **constant** signals

6. Logic expressions must be **minimal**.

Note: Here are some examples:

- If the question requires you to write the output of Register R1, then your answer must be **R1out[7:0].** If the question requires you to only write bit 6 of the output of Register R1, then your answer must be **R1out6.**
- If the question requires you to write the logic expression for ALU NAND operation, then your answer must be written as **~(Src1[7:0] AND Src2[7:0])** because you are only allowed to express logic expressions using AND, OR, NOT operations.

Now, complete the following fill-in-the-blank questions for boxes marked with magenta **color and denoted with letters marked P through X, and toM, addressM, and writeM.**

1. P = **[A]**
2. Q = **[B]**
3. R = **[C]**
4. S = **[D]**
5. T = **[E]**
6. U = **[F]**
7. V = **[G]**
8. W = **[H]**
9. X = **[I]**
10. toM = **[J]**
11. addressM = **[K]**
12. writeM = **[L]**

Specified Answer for: A ✗ x and y

Specified Answer for: B ✗ x and y

Specified Answer for: C ✗ x and y

Specified Answer for: D ✗ x and y

Specified Answer for: E ✗ x and y

Specified Answer for: F ✗ x and y

Specified Answer for: G ✗ x and y

Specified Answer for: H ✗ x or y

Specified Answer for: I ✗ x and y

Specified Answer for: J ✗ x and y

Specified Answer for: K ✗ x and y

Specified Answer for: L ✗ x and y

Correct Answers for: A

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	~(In7 AND In6)	
✓ *Exact Match*	~(In6 AND In7)	
✓ *Exact Match*	~(In6) OR ~(In7)	
✓ *Exact Match*	~(In7) OR ~(In6)	
✓ *Exact Match*	~(In[7] AND In[6])	
✓ *Exact Match*	~(In[6] AND In[7])	
✓ *Exact Match*	~(In7 AND In6)	
✓ *Exact Match*	~(In6 AND In7)	
✓ *Exact Match*	~(In6) OR ~(In7)	
✓ *Exact Match*	~(In7) OR ~(In6)	
✓ *Exact Match*	~(In[7] AND In[6])	
✓ *Exact Match*	~(In[6] AND In[7])	

Correct Answers for: B

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In5 AND ~(In7 AND ~ (In6))	
✓ *Exact Match*	In5 AND ~(In7 AND ~In6)	
✓ *Exact Match*	In5 AND (~(In7) OR In6)	
✓ *Exact Match*	In5 AND (~In7 OR In6)	
✓ *Exact Match*	In5 AND ~(In7)	
✓ *Exact Match*	~(In7) AND In5	
✓ *Exact Match*	In5 AND ~(In7 AND ~ (In6))	
✓ *Exact Match*	In5 AND ~(In7 AND ~In6)	
✓ *Exact Match*	In5 AND (~(In7) OR In6)	
✓ *Exact Match*	In5 AND (~In7 OR In6)	
✓ *Exact Match*	In5 AND ~(In7)	
✓ *Exact Match*	~(In7) AND In5	

Correct Answers for: C

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In4 AND ~(In7 AND ~(In6))	
✓ *Exact Match*	In4 AND ~(In7 AND ~In6)	
✓ *Exact Match*	In4 AND (~(In7) OR In6)	
✓ *Exact Match*	In4 AND (~In7 OR In6)	
✓ *Exact Match*	In4 AND ~(In7)	
✓ *Exact Match*	~(In7) AND In4	
✓ *Exact Match*	In4 AND ~(In7 AND ~(In6))	
✓ *Exact Match*	In4 AND ~(In7 AND ~In6)	
✓ *Exact Match*	In4 AND (~(In7) OR In6)	
✓ *Exact Match*	In4 AND (~In7 OR In6)	
✓ *Exact Match*	In4 AND ~(In7)	
✓ *Exact Match*	~(In7) AND In4	

Correct Answers for: D

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In7	
✓ *Exact Match*	In[7]	
✓ *Exact Match*	In7 AND ~(In6)	
✓ *Exact Match*	~(In6) AND In7	
✓ *Exact Match*	In7	
✓ *Exact Match*	In[7]	
✓ *Exact Match*	In7 AND ~(In6)	
✓ *Exact Match*	~(In6) AND In7	

Correct Answers for: E

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In3	
✓ *Exact Match*	In[3]	
✓ *Exact Match*	In3	
✓ *Exact Match*	In[3]	

Correct Answers for: F

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In2	
✓ *Exact Match*	In[2]	
✓ *Exact Match*	In2	
✓ *Exact Match*	In[2]	

Correct Answers for: G

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In1	
✓ *Exact Match*	In[1]	
✓ *Exact Match*	In1	
✓ *Exact Match*	In[1]	

Correct Answers for: H

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In0	
✓ *Exact Match*	In[0]	
✓ *Exact Match*	In0	
✓ *Exact Match*	In[0]	

Correct Answers for: I

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In6	
✓ *Exact Match*	In[6]	
✓ *Exact Match*	In6	
✓ *Exact Match*	In[6]	

Correct Answers for: J

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	R3out[0..7]	
✓ *Exact Match*	R3out	
✓ *Exact Match*	R3out[7..0]	
✓ *Exact Match*	R3out[7:0]	
✓ *Exact Match*	R3out[0:7]	

Correct Answers for: K

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In[0..5]	
✓ *Exact Match*	In[0:5]	
✓ *Exact Match*	In[5..0]	
✓ *Exact Match*	In[5:0]	
✓ *Exact Match*	In[0..5]	
✓ *Exact Match*	In[0:5]	
✓ *Exact Match*	In[5..0]	
✓ *Exact Match*	In[5:0]	

Correct Answers for: L

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	In7 AND In6	
✓ *Exact Match*	In6 AND In7	

✓ *Exact Match*	(In7 AND In6)
✓ *Exact Match*	(In6 AND In7)
✓ *Exact Match*	In[7] AND In[6]
✓ *Exact Match*	In[6] AND In[7]
✓ *Exact Match*	In7 AND In6
✓ *Exact Match*	In6 AND In7
✓ *Exact Match*	(In7 AND In6)
✓ *Exact Match*	(In6 AND In7)
✓ *Exact Match*	In[7] AND In[6]
✓ *Exact Match*	In[6] AND In[7]

Response Feedback: P is the input of the DMUX4Way that generates the load signal for registers. Load is true in case if it is a arith/logical/memread instruction. Therefore it will be ~(In7 AND In6), where In7 and In6 are the opcode bits.

Q and R are the select lines of the DMUX4Way that generates the load signal for registers. The selection is for the register identified as the destination register. In case of arithmetic/logical instruction, the destination is identified by I5 and I4 bits (00 for Register 0, 01 for Register 1, etc.). In case of Mem Read, the destination register is always Register 0. Hence Q = In5 AND ~(In7 AND ~(In6)). Q= In5 AND ~(In7) is also acceptable since Memory Write does not generate an active load signal anyway. R = In4 AND ~(In7 AND ~(In6)). R = In4 AND ~(In7) is also acceptable since Memory Write does not generate an active load signal anyway.

S is the select line of MUX8 that selects between READ data coming from Memory or Register Update coming from ALU. S=In7 or S=In7 AND ~(In6) are acceptable.

T and U are selects of MUX4Way8 for SRC1. These come straight from In3 and In2 respectively.

V and W are selects of MUX4Way8 for SRC2. These come straight from In1 and In0 respectively.

X specifies ALU operation NAND or ADD therefore In6 is used to differentiate between these two operations.

toM is the write data and that comes straight from Register R3 therefore toM = R3out[7:0]

addressM is in the instruction itself specified by In[5:0] for the Write Instruction

writeM is the case when In7 and In6 are 11. Therefore writeM=In7 AND In6

Question 11

1 out of 1 points

In the second pass of the HACK Computer two-pass assembly process, the assembler does not handle

Selected Answer: ✓ C. Label symbols (e.g. "(LOOP)")

Answers:

A. A-instructions that contain variable symbols (e.g. "@R0")

B. C-instructions (compute instructions, e.g. "D=M")

✓ C. Label symbols (e.g. "(LOOP)")

D. A-instructions (addressing instructions, e.g. "@11")

Question 12

1 out of 1 points

When translating D+M into binary code, the HACK assembler

Selected Answer: ✓ D.
Generates the binary equivalent of the D+M mnemonic (a mnemonic is symbolic representation of an operation)

Answers:

A. Evaluates D+M and stores the result in a temporary variable

B. Evaluates D+M and stores the result in the symbol table

C. Evaluates D, evaluates M, and emits binary code that adds them up

✓ D.
Generates the binary equivalent of the D+M mnemonic (a mnemonic is symbolic representation of an operation)

Question 13

3 out of 3 points

Based on HACK Instruction Reference (here), answer the following questions pertaining to the generation of binary machine code

Note: Binary machine code must be written without any whitespace between the bits.

What is the HACK binary machine code for @100 ? **[A]**

What is the HACK binary machine code for AM = D | M ? **[B]**

What is the HACK binary machine code for D-1 ; JGE ? **[C]**

Specified Answer for: A ✓ 0000000001100100

Specified Answer for: B ✓ 1111010101101000

Specified Answer for: C ✓ 1110001110000011

Correct Answers for: A		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	0000000001100100	
Correct Answers for: B		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	1111010101101000	
Correct Answers for: C		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	1110001110000011	

Question 14

1 out of 1 points

How would the HACK assembler distinguish branching and variable symbols?

Note: Branching is associated with Labels

Selected Answer: ✓ C.
A branching symbol has a label declaration somewhere in the program, and a variable symbol doesn't.

Answers:

A.
A variable symbol must be declared before it is used, while a branching symbol can be used before being declared.

B.
A branching symbol must be declared before it is used, while a variable symbol can be used before being declared.

✓ C.
A branching symbol has a label declaration somewhere in the program, and a variable symbol doesn't.

D.
A variable symbol has a label declaration somewhere in the program, and a branching symbol doesn't.

Question 15

2 out of 2 points

The number of bits representing the Virtual Page Number (pointer to the page table) in a virtual memory system with 8 GB Physical Memory, 256 GB Virtual Memory, and 4 MB Page Size is

Selected Answer: ✓ C. 16

Answers:

A. 4

B. 8

✓ C. 16

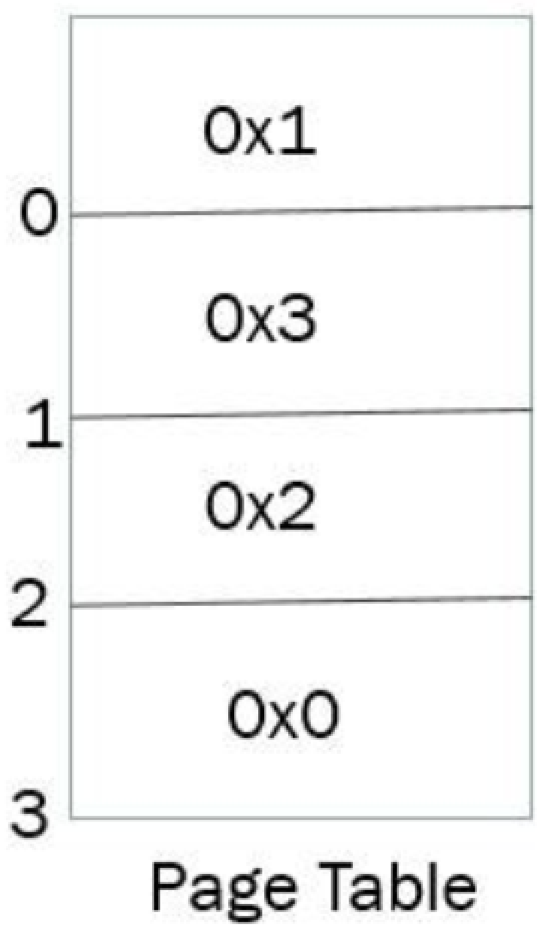
D. 12

Question 16

0 out of 5 points

Suppose we have a virtual memory system with only **4 page table entries**. The virtual memory system contains a **32-bit physical address space and a 32-bit virtual address space**. **What is the translated physical address from the following virtual address: 0x2C51908F?**

Note: The "0x" prefix is used to denote content in hexadecimal number system.



The translated physical address is [A]

Note: Write the HEX address without spaces between the HEX digits.

Selected Answer: ✗ 0x2C51908F

Correct Answer:

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	0x6C51908F	
✓ *Exact Match*	6C51908F	
✓ *Exact Match*	0x6C51908F	
✓ *Exact Match*	01101100010100011001000010001111	
✓ *Exact Match*	6C51908F	

Response Feedback:

0x2C51908F = 0010 1100 0101 0001 1001 0000 1000 1111

FROM THE ABOVE, THE VIRTUAL ADDRESS IS PARTITIONED AS FOLLOWS:

0010 1100 0101 0001 1001 0000 1000 1111
= 00 (VPN) AND 10 1100 0101 0001 1001 0000 1000 1111 (OFFSET)

PAGE TABLE ENTRY 0 CONTAINS FRAME # 0x1 i.e. 0001.

WE NOW COMBINE THE 2 LEAST SIGNIFICANT BITS WITH THE OFFSET TO GENERATE THE PHYSICAL ADDRESS AS FOLLOWS:

PHYSICAL ADDRESS = 01 CONCATENATED
WITH 10 1100 0101 0001 1001 0000 1000 1111

= 0110 1100 0101 0001 1001 0000 1000 1111 = 0x6C51908F

Question 17

4 out of 4 points

A computer system has 4 GB of Virtual Memory, 1 MB of Physical Memory, and 1 KB page size. How many page faults will result from the following 4 memory transactions:

READ 0xF0012345

WRITE 0x5432100F

READ 0x54321012

WRITE 0xF0011012

Please write the number of page faults and a brief explanation for your answer in the space provided below. Full points will only be awarded to answers with supporting explanation.

Note: the addresses are in HEXADECIMAL notation with prefix 0x. Also assume that all page table entries are assumed to be initialized as invalid.

Selected Answer:

# of page faults : 3
Brief Explanation:
Since page size is 1 kilobytes and offset = log (page size) , we can conclude that offset = log (1 KB) = 10 bits. The number of page faults is 3 because the 2nd & 3rd references on the page are identical. While, the 4th and 1st references have differing page numbers (a miss).

Correct Answer:

✓

READ F0012345: +1 page fault because the page has not been retrieved before
WRITE 5432100F: +1 because the page has not been retrieved before
READ 54321012: +0 because it is in the same page as 5432100F and is stored in physical memory
WRITE F0011012: +1 because the page has not been retrieved before

Total Page Faults: 3

Response Feedback: [None Given]

Question 18

6 out of 6 points

A cache has the following parameters: **Total Size (512 KB), Line Size (256 Bytes), 4-way set associative.** **How many bits are required for Tag, Index and Offset for a 32-bit Memory address?** Assume that the cache is byte-addressable.

Number of Tag Bits = **[A]**

Number of Set Index Bits = **[B]**

Number of Byte Offset Bits = **[C]**

RULE: Only write the numerical value of bits for the answers above. For example, if the number of tag bits is 12 then just write 12.

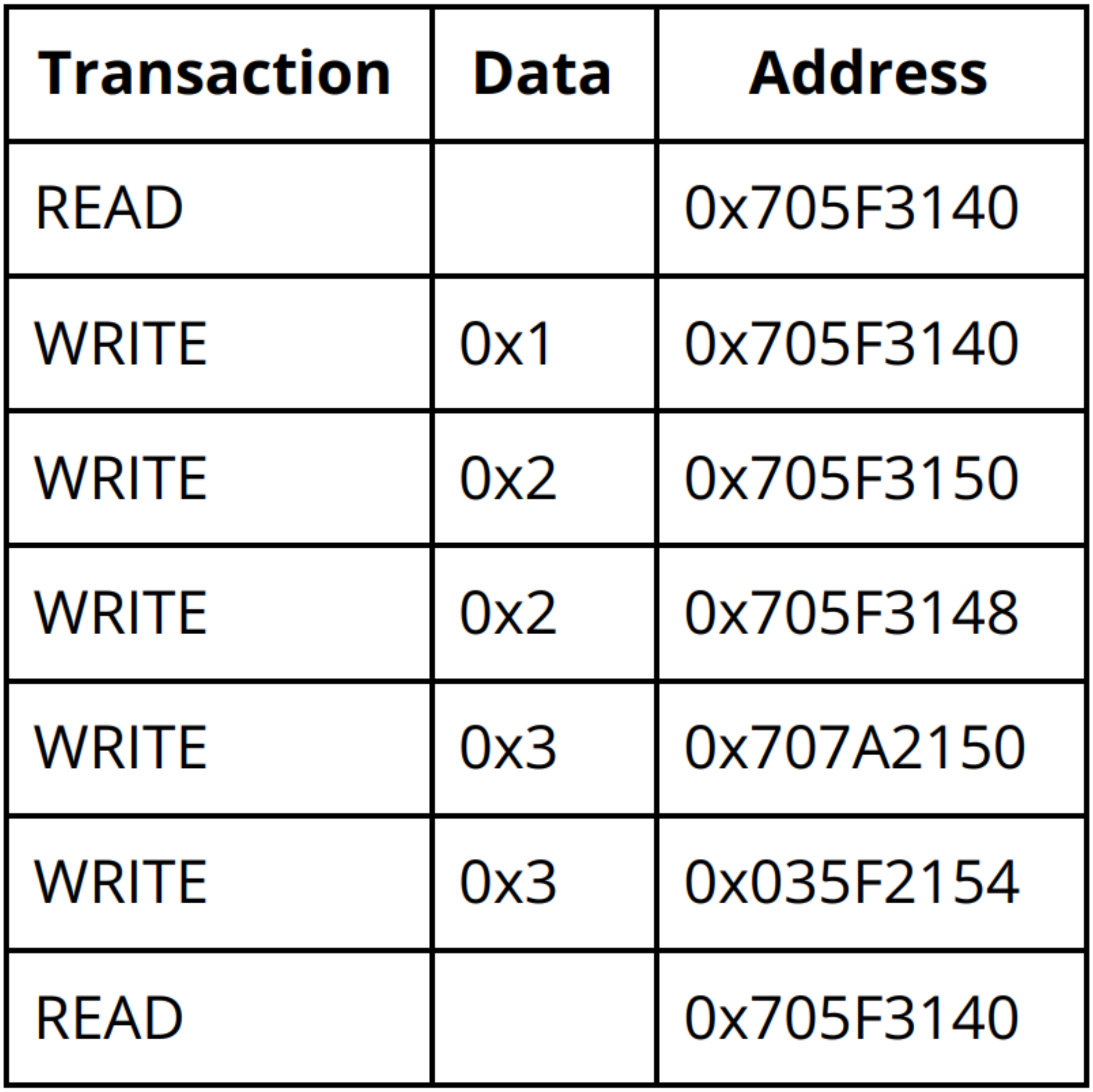
Specified Answer for: A	✓ 15
Specified Answer for: B	✓ 9
Specified Answer for: C	✓ 8

Correct Answers for: A		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	15	
✓ *Exact Match*	15 bits	
✓ *Exact Match*	15bits	
Correct Answers for: B		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	9	
✓ *Exact Match*	9 bits	
✓ *Exact Match*	9bits	
Correct Answers for: C		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	8	
✓ *Exact Match*	8 bits	
✓ *Exact Match*	8bits	

Question 19

4 out of 6 points

Assume an 8 KB cache with 32 Byte lines on a machine that uses 32-bit virtual and physical addresses. Also assume that our cache is a 2-way set associative cache that uses an LRU replacement policy. To test it out, we try the following sequence of reads and writes:



Transaction	Data	Address
READ		0x705F3140
WRITE	0x1	0x705F3140
WRITE	0x2	0x705F3150
WRITE	0x2	0x705F3148
WRITE	0x3	0x707A2150
WRITE	0x3	0x035F2154
READ		0x705F3140

Recall that the unit of transfer between cache and main memory is one block (line). ***Assume a write allocate policy for write miss. Initially, assume the cache is empty.*** **Please fill in the blanks to submit answer to the following questions:**

(a) The total number of misses in the above access pattern is **[A]**

(b) Assume the cache is a write-through cache. The total number of writes from the cache to the memory is **[B]**

(c) Assume the cache is a write-back cache. The total number of writes from the cache to the memory is **[C]**

Hint: To answer the above questions, you may want to start by first writing the translation of the above addresses into their respective TAG, INDEX, and BYTE OFFSET FIELDS next to the table above.

Specified Answer for: A ✗ 10

Specified Answer for: B ✓ 5

Specified Answer for: C ✓ 2

Correct Answers for: A

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	4	

Correct Answers for: B

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	5	

Correct Answers for: C

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	2	

Response Feedback: **MISSES**: The first READ to 0x705F3140 will generate a MISS, then WRITE 0x707A2150 will generate a MISS, then WRITE 0x035F2154 will generate a MISS and result in eviction of 0x705F3140 so finally the READ 0x705F3140 results in a MISS as well.

WRITES(Write-Through): For a write-through cache, you write to main memory every time there is a cache write so number of writes will be 5.

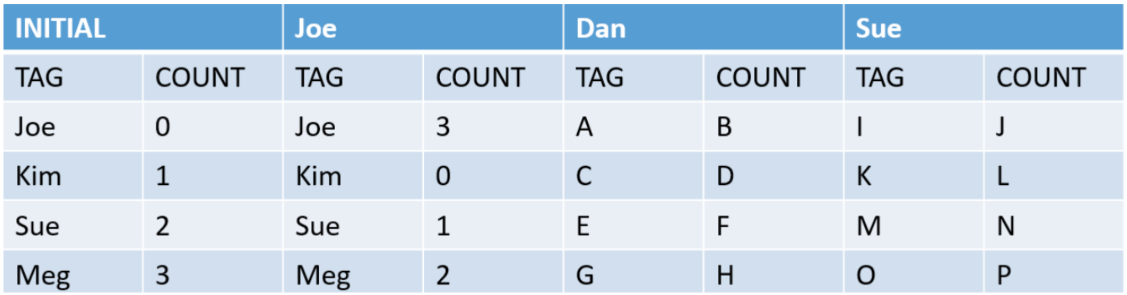
WRITES (Write-Back): For a write-back cache, you write to main memory only when a dirty line is evicted, so number of writes will be 2.

Question 20

4 out of 4 points

Assume a fully associative cache with 4 lines. Assume that the initial state of the tags are Joe, Kim, Sue, and Meg respectively as shown below. Applying the LRU policy, complete the table below for the following tag address sequence: Joe -> Dan -> Sue

You can complete the table by replacing labels noted from A through P. Note that the TAG will be the names and COUNT will be numbers. The first column "Joe" is completed for you as a reference.



INITIAL		Joe		Dan		Sue	
TAG	COUNT	TAG	COUNT	TAG	COUNT	TAG	COUNT
Joe	0	Joe	3	A	B	I	J
Kim	1	Kim	0	C	D	K	L
Sue	2	Sue	1	E	F	M	N
Meg	3	Meg	2	G	H	O	P

A=**[A]** B=**[B]** I=**[I]** J=**[J]**

C=**[C]** D=**[D]** K=**[K]** L=**[L]**

E=**[E]** F=**[F]** M=**[M]** N=**[N]**

G=**[G]** H=**[H]** O=**[O]** P=**[P]**

Specified Answer for: A	✓ Joe
Specified Answer for: B	✓ 2
Specified Answer for: I	✓ Joe
Specified Answer for: J	✓ 1
Specified Answer for: C	✓ Dan
Specified Answer for: D	✓ 3
Specified Answer for: K	✓ Dan
Specified Answer for: L	✓ 2
Specified Answer for: E	✓ Sue
Specified Answer for: F	✓ 0
Specified Answer for: M	✓ Sue
Specified Answer for: N	✓ 3
Specified Answer for: G	✓ Meg
Specified Answer for: H	✓ 1
Specified Answer for: O	✓ Meg
Specified Answer for: P	✓ 0

Correct Answers for: A

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	Joe	

Correct Answers for: B

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	2	

Correct Answers for: I

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	Joe	

Correct Answers for: J

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	1	

Correct Answers for: C

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	Dan	

Correct Answers for: D

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	3	

Correct Answers for: K

Evaluation Method	Correct Answer	Case Sensitivity

✓ *Exact Match*	Dan	
Correct Answers for: L		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	2	
Correct Answers for: E		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	Sue	
Correct Answers for: F		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	0	
Correct Answers for: M		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	Sue	
Correct Answers for: N		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	3	
Correct Answers for: G		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	Meg	
Correct Answers for: H		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	1	
Correct Answers for: O		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	Meg	
Correct Answers for: P		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	0	

Question 21

0 out of 4 points

Consider the following processors (ns stands for nanoseconds):

P1: 4-stage pipeline with stage delays 1 ns, 2 ns, 2 ns, 1 ns.
P2: 4-stage pipeline with stage delays 1 ns, 1.5 ns, 1.5 ns, 1.5 ns.
P3: 5-stage pipeline with stage delays 0.5 ns, 1 ns, 1 ns, 0.6 ns, 1 ns.
P4: 5-stage pipeline with stage delays 0.5 ns, 0.5 ns, 1 ns, 1 ns, 1.1 ns.

The processor with the highest clock frequency is

Selected Answer: ✗ D. P4

Answers: A. P2

✓ B. P3

C. P1

D. P4

Response Feedback: The slowest stage sets the frequency. P3's slowest stage has the best delay of all the processors.

Question 22

3 out of 6 points

The following code fragment (instructions L1:L4) is executed in a 5-stage pipeline with stages Fetch(**F**) , Decode/Read Register (**D**) , Execute (**E**) , Memory (**M**) , and Writeback Register (**W**). Please examine the instruction dependencies closely to understand the pipelined execution of the code fragment. Then answer the following four questions:

L1: R1 = R2 + R3

L2: R2 = MEM[R4]

L3: R4 = R1 + R3

L4: MEM[R5] = R3

	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**	**C13**	**C14**	**C15**
L1															
L2															
L3															
L4															

Note: For each question below, write the cycle (for example: write C5 if a given instruction is completing in cycle C5).

(A) Instruction L1 will complete in cycle [A]

(B) Instruction L2 will complete in cycle [B]

(C) Instruction L3 will complete in cycle [C]

(D) Instruction L4 will complete in cycle [D]

Specified Answer for: A ✓ C5

Specified Answer for: B ✓ C6

Specified Answer for: C ✗ C9

Specified Answer for: D ✗ C10

Correct Answers for: A

Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	C5	
✓ *Exact Match*	5	

Correct Answers for: B

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	C6	
✓ *Exact Match*	6	

Correct Answers for: C

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	C8	
✓ *Exact Match*	8	

Correct Answers for: D

Evaluation Method	Correct Answer	Case Sensitivity
✓ *Exact Match*	C9	
✓ *Exact Match*	9	

Tuesday, May 11, 2021 11:36:17 AM CDT

← **OK**