CMPE109 FUNDAMENTALS OF COMPUTING
2024-2025 FALL
LAB ASSIGNMENT 3
SECTION 2
A

A

Full Name:

Student ID:

Signature:

Q1) Trace the following assembly code. After we run the code, whether it halts or gives an error (doesn't matter), what will be the hexadecimal values stored in the following addresses, including IR. **[5 points]**

Note: Every hexadecimal value you write must be 2 characters. For example, write 0A for decimal 10.

Machine Instruction		Assembly Instruction	Operation
Op-code	Operand		
1	RXY	load R, [XY]	Load R with the content from the memory cell at address XY
2	RXY	load R, XY	Load R with the bit pattern XY
3	RXY	store R, [XY]	Store the content of R into the memory cell at address XY
4	0RS	move S, R	Move content of R into S
5	RST	addi R, S, T	Add S and T and put the result in R (R, S, and T are in two's complement integer notation)
6	RST	addf R, S, T	Add S and T and put the result in R (R, S, and T are in floating-point notation)
7	RST	or R, S, T	OR the bit patterns in S and T and put the result in R
8	RST	and R, S, T	AND the bit patterns in S and T and put the result in R
9	RST	xor R, S, T	XOR the bit patterns in S and T and put the result in R
A	R0X	ror R, X	Circularly rotate the bit pattern in R one bit to the right X times
B	RXY	jmpEQ R=R0, XY	Start decoding the instruction located at address XY if the bit pattern in R is equal to the bit pattern in register 0
C	000	halt	Halt execution
D	0RS	load R, [S]	Load R with the content from the memory cell whose address is in S
E	0RS	store R, [S]	Store the content of R into the memory cell whose address is in S
F	RXY	jmpLE R<=R0, XY	Start decoding the instruction located at address XY if the bit pattern in R is less than or equal to the bit pattern in register 0

```
1          LOAD R0, 10
2          LOAD R1, 0
3          LOAD R2, 2
4          LOAD R3, LIST
5
6   BACK:  LOAD R4, [R3]
7          JMPEQ R4=R0, END
8          ADDI R1, R1, R4
9          ADDI R3, R3, R2
10         STORE R4, [$FE]
11         JMP BACK
12
13  END:   ROR R1, 2
14         STORE R1, [$FF]
15         ROR R4, 1
16         HALT
17
18  LIST:  DB 2,8,5,10,9,9,10,4,-1
```

R4	
FF	
R1	

Q2) Above assembly code can run without giving and error, or maybe it is wrong and can give an error at some point, you must understand that. Do you think this assembly code will run and halt without an error (write **yes** or **no**)? If you said **no**, why (write just 1 sentence)? **[2 points]**

Q3) Recreate the given webpage visually using HTML. Your code should include a nested list about hamburger ingredients. Ensure your code is correctly structured with proper HTML tags and formatting. **[7 points]**

```
<!DOCTYPE html>
<html>
<head>


</head>
<body                              >









</body>
</html>
```

Cute Hamburger Ingredients

- *Bun UwU*
  1. Flour
  2. Water **(I love it!)**
  3. Sugar
  4. Salt **(Yuck!)**
  5. Yeast mixture
- *Patty OwO*
  1. Ground meat
  2. Onion **(Smelly!)**
  3. Salt
  4. Black pepper
  5. Spices
- *General Ingredients :3*
  - Lettuce
  - Tomato
  - Cheese
  - Ketchup
  - Mayonnaise
  - Pickles **(No!)**

- Page title is “My Recipes”.
- Page background is colored with “#FFCCFF” color.
- Use “h1” for the heading.
- Image is placed <u>after the heading</u>, its name is “brgA.png”, width is “168”, height is “300”, and it is aligned to the “right”. It looks on the right of the list because it is <u>aligned to the right</u>, there is no special case or code for that.
- Outer list items are <u>*italic*</u>.
- Parentheses in inner list items are <u>**bold**</u>.

A