

Qno.1)

Code:

```
ORG 100
    Load HChar  // Load 'H' into the accumulator
    Output

    Load EChar  // Load 'e' into the accumulator
    Output

    Load LChar  // Load 'l' into the accumulator
    Output

    Load LChar  // Load 'l' into the accumulator
    Output

    Load OChar  // Load 'o' into the accumulator
    Output

    Halt

HChar, HEX 48  // ASCII for 'H'
EChar, HEX 65  // ASCII for 'e'
LChar, HEX 6C  // ASCII for 'l'
OChar, HEX 6F  // ASCII for 'o'
```

Output:

The screenshot shows the MARIE.js web simulator interface. The assembly code is as follows:

```

1  ORG 100
2  Load HChar // Load 'H' into the accumulator
3  Output
4
5  Load EChar // Load 'e' into the accumulator
6  Output
7
8  Load LChar // Load 'l' into the accumulator
9  Output
10
11 Load LChar // Load 'l' into the accumulator
12 Output
13
14 Load OChar // Load 'o' into the accumulator
15 Output
16
17 Halt
18

```

On the right, the register values are displayed:

- AC: 006F
- IR: 7000
- MAR: 10A
- MBR: 7000
- PC: 10B
- IN: 0000
- OUT: 006F

The output list shows the text "Hello". Below the code, a memory dump table is visible, showing hexadecimal values for addresses from 000 to 090. At the bottom, there are control buttons: Assemble, Step, Microstep, Step Back, Halted, Restart, and a Delay slider set to 1 ms.

qno.2)

ORG 100

LOAD ZERO // Initialize i to 0
STORE I

LOOP, LOAD I // Load i
SUBT THREE // Subtract 3 from i
SKIPCOND 000 // If result is Zero (i.e., i == 3), skip next line
JUMP PRINT // If i is not 3, jump to print
JUMP INCREMENT // If i is 3, jump to increment

PRINT, OUTPUT

INCREMENT, LOAD I // Reload i (necessary after OUTPUT and to handle 'continue')
ADD ONE // Increment i
STORE I // Store the new value of i

LOAD FIVE // Load the loop end condition (5)
SUBT I // Subtract i from 5

```

SKIPCOND 000    // If result is Negative (i.e., i >= 5), skip next line
JUMP LOOP      // If i is less than 5, jump back to the start of the loop

```

ENDLOOP, HALT

```

I, DEC 0        // Memory location for loop variable i
ONE, DEC 1      // Constant 1
THREE, DEC 3    // Constant 3 (condition for continue)
FIVE, DEC 5     // Constant 5 (loop end condition)
ZERO, DEC 0     // Constant 0

```

Output:

The screenshot shows the MARIE.js web application interface. The top part displays the assembly code being executed. The middle part shows the machine state, including registers (AC, IR, MAR, MBR, PC, IN, OUT) and their current values. The bottom part shows a memory dump table with addresses and hexadecimal values. At the very bottom, there are control buttons for execution and a delay slider.

Assembly code:

```

1  ORG 100
2  LOAD ZERO      // Initialize i to 0
3  STORE I
4
5  LOOP, LOAD I    // Load i
6  SUBT THREE     // Subtract 3 from i
7  SKIPCOND 000   // If result is Zero (i.e., i == 3), skip next line
8  JUMP PRINT     // If i is not 3, jump to print
9  JUMP INCREMENT // If i is 3, jump to increment
10
11 PRINT, OUTPUT
12
13 INCREMENT, LOAD I // Reload i (necessary after OUTPUT and to handle 'continue')
14 ADD ONE          // Increment i
15 STORE I          // Store the new value of i
16
17 LOAD FIVE        // Load the loop end condition (5)
18 SUBT I           // Subtract i from 5

```

Machine state:

- AC: FFFF
- IR: 7000
- MAR: 10F
- MBR: 7000
- PC: 110
- IN: 0000
- OUT: 0002

Machine halted normally.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Control buttons: Assemble, Step, Microstep, Step Back, Halted, Restart, Delay: 1 ms

Qno.3)

ORG 100

LOAD ZERO

STORE I

LOOP, LOAD I

SUBT THREE

SKIPCOND 400

JUMP SKIP_PRINT

JUMP INCREMENT

SKIP_PRINT, LOAD I

OUTPUT

INCREMENT, LOAD I

ADD ONE

STORE I

LOAD FIVE

SUBT I

SKIPCOND 400

JUMP LOOP

END_LOOP, HALT

I, HEX 0

ONE, DEC 1

THREE, DEC 3

FIVE, DEC 5

ZERO, HEX 0

Output:

MARIE.js | Home

Untitled document - Google

marie.js.org

Assembly code:

```

1  ORG 100
2
3  LOAD ZERO
4  STORE I
5
6  LOOP, LOAD I
7  SUBT THREE
8  SKIPCOND 400
9  JUMP SKIP_PRINT
10 JUMP INCREMENT
11
12 SKIP_PRINT, LOAD I
13 OUTPUT
14
15 INCREMENT, LOAD I
16 ADD ONE
17 STORE I
18

```

Machine halted normally.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

AC 0000

IR 7000

MAR 110

MBR 7000

PC 111

IN 0000

OUT 0004

OUTPUT MODE: DEC

Assemble Step Microstep Step Back Halted Restart Delay: 1 ms

Qno.4)

ORG 100

Input, DEC 0
 JUMP InputCheck

InputCheck, DEC 0
 SUBT ZeroCheck
 JUMP Error
 JUMP InputN

InputN, DEC 0
 JUMP InputCheckN

```
InputCheckN, DEC 0
            SUBT ZeroCheck
            JUMP Error
            JUMP Initialize
```

```
Error,      JUMP Halt
```

```
ErrorMsg,   DEC -1
```

```
Initialize, STORE M
            STORE N
            CLEAR
            JUMP Loop
```

```
Loop,       ADD M
            SUBT One
            SUBT ZeroCheck
            JUMP Loop
```

```
Halt,       HALT
```

```
One,        DEC 1
```

```
M,          DEC 0
N,          DEC 0
```

```
ZeroCheck,  DEC 0
```

```
            END Start
```

Output:

MARIE.js | Home

Untitled document - Google

marie.js.org

Relaunch to update

HomeFileExamplesEditViewHelp

Assembly code:

Autosaved file

AC

0000

IR

0000

MAR

000

MBR

0002

PC

001

IN

0000

OUT

0000

Output log

RTL log

Watch list

Input list

OUTPUT MODE:

DEC

1

ORG 100

2

3

Input,

DEC 0

4

JUMP InputCheck

5

6

InputCheck,

DEC 0

7

SUBT ZeroCheck

8

JUMP Error

9

JUMP InputN

10

11

InputN,

DEC 0

12

JUMP InputCheckN

13

14

InputCheckN,

DEC 0

15

SUBT ZeroCheck

16

JUMP Error

17

JUMP Initialize

18

Running...

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	0002	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Assemble

Step

Microstep

Step Back

Pause

Restart

Delay: 1 ms