1. Given code in assembly

.global \_start

\_start: br Start  # begin at the main program

# Constants

    .equ HEX\_DISP, 0x88A0

    .equ HEX\_CONT, 0x88B0

.org 0x0100

Start:

# r2 is the counter

    init:

        # enable all hex displays

        movia r3, HEX\_CONT

        movi r4, 0xFF  # enable all hex displays

        stw r4, 0(r3)  # store to HEX\_CONT

        movia r5, HEX\_DISP  # r5 points to the first hex display

        movi r2, 0  # initialize counter to 0

    loop:

        stw r2, 0(r5)  # store the counter value to the hex display

        addi r2, r2, 1  # increment the counter

        br loop  # repeat the loop

1. Subroutines
2. void outchar(char ch){
3. volatile int \*const UART = (int \*)0x00008840; // UART register
4. \*UART = ch; // send character to UART
5. }
6. char bin2hex(char N){
7. // take 4 LSB of N, convert to hex, return
8. N &= 0x0F; // mask to get 4 LSB
9. if (N < 10) {
10. printf("bin2hex: Output = %c\n", N + 0x30); // Debug print
11. return N + '0'; // convert to ASCII for digits 0-9
12. }
13. else {
14. printf("bin2hex: Output = %c\n", N + 0x37); // Debug print
15. return N - 10 + 'A'; // convert to ASCII for letters A-F
16. }
17. }
18. void outhex(char N){
19. outchar(bin2hex((N & 0xF0) >> 4)); // send first hex digit
20. outchar(bin2hex(N & 0x0F)); // send first hex digit
21. }

The meaning of void in front of outhex(char N) means it doesn’t return a value from the subroutine.