

---

**Assignment #5**  
**Assembly Language Programs II**

---

1. For this problem assume the memory/register contents are as shown in the following tables

Location	Value
W0	0x2004
W1	0x2006
W2	0xAFDE
W3	0x4FD2
W4	0x0006
W15	0x2006

Location	Value
0x2000	0xECDC
0x2002	0x38F2
0x2004	0x0044
0x2006	0xBC4D
0x2008	0xFFFF
0x200A	0x2523
0x200C	0xD23A
0x200E	0x98C3

at the *start* of each instruction. Give the modified memory location or register and its content.

- (a) `mov [W1], W0`
  - (b) `mov W1, [W0+W4]`
  - (c) `mov mov W2, [W0++]`
  - (d) `mov.b W2, [--W1]`
  - (e) `mov.b [W1+W4], W2`
  - (f) `mov [W0-4], W1`
  - (g) `add [W0++], [--W1], W3`
  - (h) `push W1`
  - (i) `pop W3`
2. Rewrite your program for counting bits (Assignment 4, number 2) as a subroutine so that it works with the following code

```
.bss
numbits1: .space 2
numbits2: .space 2
numdone: .space 2

.text
```

```

.global __reset
__reset
    mov #2, W1
    mov #0x4FC2, W0
    call countbits
    mov W0, numbits1
    mov #F257
    call countbits
    mov W0, numbits2
    mov W1, numdone
done:
    goto done

```

In addition to the source code, provide a printout of the correct values stored in the file registers.

3. Write an assembly language subroutine that receives as input parameters 1) a pointer to an array in W0 and 2) the number of unsigned integers in the array in the register W1. The routine returns the maximum number found in the array in W0. Make sure you preserve all register values (besides W0). Also write a “main” routine that tests the functionality of the subroutine with at least a 20 element array. Provide a print out of the source code and whatever is needed to show the functionality is correct.