Homework:

Question 1:

Given code:

```
    li $t8, -39
    srav $v0, $t1, $t8
    lw $t8, -52($a2)
```

Expand macros:

```
    ori $t8, $0, -39
    srav $v0, $t1, $t8
    lw $t8, -52($a2)
```

Convert register names to register numbers:

```
    ori $24, $0, -39
    srav $2, $9, $24
    lw $24, -52($4)
```

Assemble the code (convert to machine code):

```
# 0011 01ss ssst tttt iiii iiii iiii iiii
# 0011 0100 0001 1000 1111 1111 1101 1001
    ori $24, $0, -39

# 0000 00ss ssst tttt dddd d000 0000 0111
# 0000 0011 0000 1001 0001 0000 0000 0111
    srav $2, $9, $24

# 1000 11ss ssst tttt iiii iiii iiii iiii
# 1000 1100 1001 1000 1111 1111 1100 1100
    lw $24, -52($4)
```

```
- address -                - machine code (binary) -                   - machine code (Hex) -

0x0040 894C     |       0011 0100 0001 1000 1111 1111 1101 1001            0x3418 FFD9
0x0040 8950     |       0000 0011 0000 1001 0001 0000 0000 0111            0x0309 1007
0x0040 8954     |       1000 1100 1001 1000 1111 1111 1100 1100            0x8C98 FFCC
```

Question 2:

instructions at address 0x0040 B80C and data start at address 0x1000 40AC

Given code:

```
    .data
value:  .word   -9
```

```
    .text
li $t4, 0

matmulv_loop:
    blez $t2, matmulv_loop_exit

    lw $t5, 0($t1)
    lw $t6, 0($t3)

    mul $t7, $t5, $t6
    add $t4, $t4, $t7

    addi $t2, $t2, -1
    addiu $t1, $t1, 4
    addu $t9, $t9, $t3

    b matmulv_loop

matmulv_loop_exit:
    sw $t4, 20($sp)
```

1) Expand macros:

```
    .data
value:  .word   -9

    .text
ori $t4, $0, 0

matmulv_loop:
    blez $t2, matmulv_loop_exit

    lw $t5, 0($t1)
    lw $t6, 0($t3)

    mult $t5, $t6
    mflo $t7

    add $t4, $t4, $t7

    addi $t2, $t2, -1
    addiu $t1, $t1, 4
    addu $t9, $t9, $t3

    bgez $0, matmulv_loop
```

```
matmulv_loop_exit:
    sw $t4, 20($sp)
```

2) convert register names to register numbers:

```
    .data
value:  .word   -9

    .text
ori $12, $0, 0

matmulv_loop:
    blez $10, matmulv_loop_exit

    lw $13, 0($9)
    lw $14, 0($11)

    mult $13, $14
    mflo $15

    add $12, $12, $15

    addi $10, $10, -1
    addiu $9, $9, 4
    addu $25, $25, $11

    bgez $0, matmulv_loop

matmulv_loop_exit:
    sw $12, 20($29)
```

3) align the labels with assembly code:

```
    .data
value:  .word   -9

    .text
ori $12, $0, 0

matmulv_loop: blez $10, matmulv_loop_exit

    lw $13, 0($9)
    lw $14, 0($11)

    mult $13, $14
```

```
        mflo $15

        add $12, $12, $15

        addi $10, $10, -1
        addiu $9, $9, 4
        addu $25, $25, $11

        bgez $0, matmulv_loop

    matmulv_loop_exit: sw $12, 20($29)
```

4) convert labels to addresses:

```
            .data
    value:  .word   -9

        .text
    ori $12, $0, 0

##################################
#   distance = +10
#   offset = 10 - 1 = 9
##################################
    matmulv_loop: blez $10, matmulv_loop_exit

        lw $13, 0($9)
        lw $14, 0($11)

        mult $13, $14
        mflo $15

        add $12, $12, $15

        addi $10, $10, -1
        addiu $9, $9, 4
        addu $25, $25, $11
##################################
#   distance = -9
#   offset = -9 - 1 = -10
##################################
        bgez $0, matmulv_loop

    matmulv_loop_exit: sw $12, 20($29)
    5) convert instructions to machine code:

            .data
    value:  .word   -9

        .text

# 0011 01ss ssst tttt iiii iiii iiii iiii
# 0011 0100 0000 1100 0000 0000 0000 0000
    ori $12, $0, 0                              <-- 0x0040 B80C

##################################
#   distance = +10
#   offset = 10 - 1 = 9
##################################

# 0001 10ss sss0 0000 iiii iiii iiii iiii
# 0001 1001 0100 0000 0000 0000 0000 1001
    matmulv_loop: blez $10, matmulv_loop_exit   <-- 0x0040 B810

# 1000 11ss ssst tttt iiii iiii iiii iiii
# 1000 1101 0010 1101 0000 0000 0000 0000
        lw $13, 0($9)                           <-- 0x0040 B814

# 1000 11ss ssst tttt iiii iiii iiii iiii
# 1000 1101 0110 1110 0000 0000 0000 0000
        lw $14, 0($11)                          <-- 0x0040 B818

# 0000 00ss ssst tttt 0000 0000 0001 1000
# 0000 0001 1010 1110 0000 0000 0001 1000
        mult $13, $14                           <-- 0x0040 B81C

# 0000 0000 0000 0000 dddd d000 0001 0010
# 0000 0000 0000 0000 0111 1000 0001 0010
        mflo $15                                <-- 0x0040 B820

# 0000 00ss ssst tttt dddd d000 0010 0000
# 0000 0001 1000 1111 0110 0000 0010 0000
        add $12, $12, $15                       <-- 0x0040 B824

# 0010 00ss ssst tttt iiii iiii iiii iiii
# 0010 0001 1000 1100 1111 1111 1111 1111
        addi $10, $10, -1                       <-- 0x0040 B828

# 0010 01ss ssst tttt iiii iiii iiii iiii
# 0010 0101 0010 1001 0000 0000 0000 0100
        addiu $9, $9, 4                         <-- 0x0040 B82C

# 0000 00ss ssst tttt dddd d000 0010 0001
# 0000 0011 0010 1011 1100 1000 0010 0001
        addu $25, $25, $11                     <-- 0x0040 B830
##################################
#   distance = -9
#   offset = -9 - 1 = -10
##################################

# 0000 01ss sss0 0001 iiii iiii iiii iiii
# 0000 0100 0000 0001 1111 1111 1111 0110
        bgez $0, matmulv_loop                  <-- 0x0040 B834

# 1010 11ss ssst tttt iiii iiii iiii iiii
# 1010 1111 1010 1100 0000 0000 0001 0100
    matmulv_loop_exit: sw $12, 20($29)         <-- 0x0040 B838
```

6) summary:

- address -	- machine code (binary) -	- machine code (Hex) -
0x0040 B80C	0011 0100 0000 1100 0000 0000 0000 0000	0x340C 0000
0x0040 B810	0001 1001 0100 0000 0000 0000 0000 1001	0x1940 0009
0x0040 B814	1000 1101 0010 1101 0000 0000 0000 0000	0x8D2D 0000
0x0040 B818	1000 1101 0110 1110 0000 0000 0000 0000	0x8D6E 0000
0x0040 B81C	0000 0001 1010 1110 0000 0000 0001 1000	0x01AE 0018
0x0040 B820	0000 0000 0000 0000 0111 1000 0001 0010	0x0000 7812
0x0040 B824	0000 0001 1000 1111 0110 0000 0010 0000	0x018F 6020
0x0040 B828	0010 0001 1000 1100 1111 1111 1111 1111	0x218C FFFF
0x0040 B82C	0010 0101 0010 1001 0000 0000 0000 0100	0x2529 0004
0x0040 B830	0000 0011 0010 1011 1100 1000 0010 0001	0x032B C821
0x0040 B834	0000 0100 0000 0001 1111 1111 1111 0110	0x0401 FFF6
0x0040 B838	1010 1111 1010 1100 0000 0000 0001 0100	0xAFAC 0014

Question 3:

```
    - address -         - binary -                             - assembly Language (Include labels if needed) -
0x0040 0000     0001 1001 1010 0000 0000 0000 0000 0011     --> blez $s, offset
                                                                0001 10ss sss0 0000 iiii iiii iiii iiii

                                                                s = 01101 = 13
                                                                offset = 0000 0000 0000 0011 = 3

                                                                formula: offset = distance - 1
                                                                distance = offset + 1 ==> 3 + 1 = 4

                                                            --> blez $13, 0x0040 0010 :-> blez $t5, 0x0040 0010 <=> blez $t5, label
```

```
0x0040 0004    0000 0001 0000 0010 0100 0000 0010 0000    --> add $d, $s, $t
                                                               0000 00ss ssst tttt dddd d000 0010 0000

                                                               s = 01000 = 8
                                                               t = 00010 = 2
                                                               d = 01000 = 8

                                                           --> add $8, $8, $2 :-> add $t0, $t0, $v0

0x0040 0008    0010 0001 0010 1010 0000 0000 0000 0001    --> addi $t, $s, immediate
                                                               0010 00ss ssst tttt iiii iiii iiii iiii

                                                               s = 01001 = 9
                                                               t = 01010 = 10
                                                               immediate = 1

                                                               addi $10, $9, immediate :-> addi $t2, $t1, 1

0x0040 000c    0000 0100 0000 0001 0000 0000 0000 0100    --> bgez $s, offset
                                                               0000 01ss sss0 0001 iiii iiii iiii iiii

                                                               s = 00000 = 0
                                                               offset = 0000 0000 0000 0100 = 4

                                                               formula: offset = distance - 1
                                                               distance = offset + 1 ==> 4 + 1 = 5

                                                           --> bgez $0, 0x0040 0020 :-> bgez $zero, offset

0x0040 0010    1010 1111 1010 1000 1111 1111 1111 0100    --> sw $t, offset($s)
                                                               1010 11ss ssst tttt iiii iiii iiii iiii

                                                               s = 11101 = 29
                                                               t = 01000 = 8
                                                               offset = 1111 1111 1111 0100 = -12

                                                           --> label: sw $t, offset($s) :-> sw $t0, -12($sp)

0x0040 0014
0x0040 0018
0x0040 001C
0x0040 0020
```