Final exam programming prep:

1- assemble the following code assuming .text starts at 0x0100 4000

```
        .text
    main:
        li $t0, -1024
        abs $t1, $t0
        add $t2, $t0, $t1

        b main
```

original code:

```
        .text
    main:
        li $t0, -1024       #  1024 = in binary => 0000010000000000
                            # -1024 = in binary => 1111101111111111
                            #                    +               1
                            #                     -----------------
                            #                      1111110000000000 = in hex => 0xFC00
        abs $t1, $t0
        add $t2, $t0, $t1

        b main
```

expand the macro instructions:

```
        .text
    main:
        lui $at, 0xFFFF
        ori $t0, $at, 0xFC00
        addu $t1, $0, $t0
        bgez $t0, 1
        sub $t1, $0, $t0
        add $t2, $t0, $t1
        bgez $0, main
```

convert register names to register numbers:

```
        .text
    main:
        lui $1, 0xFFFF
        ori $8, $1, 0xFC00
        addu $9, $0, $8
```

```
        bgez $8, 1
        sub $9, $0, $8
        add $10, $8, $9
        bgez $0, main

align labels with assmebly code:
        .text
    main: lui $1, 0xFFFF
        ori $8, $1, 0xFC00
        addu $9, $0, $8
        bgez $8, 1
        sub $9, $0, $8
        add $10, $8, $9
        bgez $0, main

convert labels to addresses:
        .text
    main: lui $1, 0xFFFF          | 0x0100 4000
        ori $8, $1, 0xFC00        | 0x0100 4004
        addu $9, $0, $8           | 0x0100 4008
        bgez $8, 1                | 0x0100 400C
        sub $9, $0, $8            | 0x0100 4010
        add $10, $8, $9           | 0x0100 4014
        bgez $0, main             | 0x0100 4018

calculate branch offsets:
        .text
    main: lui $1, 0xFFFF          | 0x0100 4000
        ori $8, $1, 0xFC00        | 0x0100 4004
        addu $9, $0, $8           | 0x0100 4008
        bgez $8, 1                | 0x0100 400C
        sub $9, $0, $8            | 0x0100 4010
        add $10, $8, $9           | 0x0100 4014
        bgez $0, main             | 0x0100 4018
        #
        # distance = -6
        # offset = distance - 1 = -6 - 1 = - 7 (or 1001 in binary 2's complement)
        #

convert instructucions to machine code:
        .text
        0011 1100 0000 0001 1111 1111 1111 1111
```

```
 main: lui $1, 0xFFFF          | 0x0100 4000

      0011 0100 0010 1000 1111 1100 0000 0000
      ori $8, $1, 0xFC00       | 0x0100 4004

      0000 0000 0000 1000 0100 1000 0010 0001
      addu $9, $0, $8          | 0x0100 4008

      0000 0101 0010 0001 0000 0000 0000 0001
      bgez $8, 1               | 0x0100 400C

      0000 0000 0000 1000 0100 1000 0010 0010
      sub $9, $0, $8           | 0x0100 4010

      0000 0001 0010 1001 0101 0000 0010 0000
      add $10, $8, $9          | 0x0100 4014

      0000 0100 0000 0001 1111 11111 11111 1001
      bgez $0, main            | 0x0100 4018
      #
      # distance = -6
      # offset = distance - 1 = -6 - 1 = -7 (or 1001 in binary 2's complement)
      #
```

2. disassemble the following code:

```
      0011 0100 0000 0010 0000 0000 0000 1010
      0000 0000 0000 0000 0000 0000 0000 1100
```

--> solution:

```
      ori $v0, $0, 10
      syscall
```

--> looking up macros:

```
     li $v0, 10
     syscall
```