James Root

ECE 3058

Lab #0

1/19/2021

1.)

X	Y	Z	RWE	IMEN	IMVA	AUEN	-A/S	LUEN	IF	SUEN	ST	LDEN	STEN	R/-W	MSEL	COMMENT
X	X	1	1	1	100	0	X	0	XXXX	0	XX	1	1	X	0	R1 = 100
1	X	1	1	0	X	0	X	0	XXXX	0	XX	1	0	0	1	R1 = M[100)
1	X	1	1	1	2	0	X	0	XXXX	1	00	0	0	X	0	R1 = 4 * R1
1	2	1	1	0	X	1	0	0	XXXX	0	XX	0	0	X	0	R1 += R2
1	2	1	1	0	X	1	0	0	XXXX	0	XX	0	0	X	0	R1 += R2
1	2	1	1	0	X	1	0	0	XXXX	0	XX	0	0	X	0	R1 = R2 + R1 (REPEAT 3X FOR += 3R2)

2.)

```
.data
# This is the start of the original array.
Original: .word 200, 270, 250, 100
      .word 205, 230, 105, 235
      .word 190, 95, 90, 205
      .word 80, 205, 110, 215
# The next statement allocates room for the other array.
# The array takes up 4*16=64 bytes.
Second: .space 64
.align 2
.globl main
.text

main: # Your fully commented program starts here.
      li $v0, 0   #store the index for original (increment by 4)
      li $v1, 0   #store the index fir Second (increment by 16)
loop:
      lw $t0, Original($v0)   #load value from first matrix
      sw $t0, Second($v1)     #store value in 2nd matrix
      addi $v0, $v0, 4        #increment to next value in Original
      addi $v1, $v1, 16       #increment to next value in Second
      slti $t1, $v1, 61       #check to see if $2 exceeds Second's size (64)
      bne $t1, $zero, check   #if $2 is not about to overflow, jump to 'check'
      addi $v1, $v1, -60      #if $2 overflows, shift to next column
check:
      slti $t2, $v0, 61       #check if $1 overflows size
      bne $t2, $zero, loop    #if not, jump to top of loop
      j Exit                  #if so, Exit the program
Exit:
      li $v0, 10 #terminate program
      syscall
```