Problem PA-2 (3 parts) Function and Stack

Part A The program `Chroma` calls a subroutine `Mystery`. Complete the program `Chroma` by adding MIPS code to correctly preserve appropriate registers before the `jal` by pushing them on the stack and to restore them after the subroutine call. Assume `Mystery` can modify any registers and that `A`, `B`, and `C` are globally defined arrays.

Chroma:	addi $1, $0, 256	# init index
Loop:	addi $1, $1, -4	# decrement index
	lw $2, A($1)	# load in current element of A
	lw $3, B($1)	# load in current element of B
	addi $29, $29, -16 **sw $1, 12($29)** **sw $2, 8($29)** **sw $3, 4($29)** **sw $31, 0($29)**	# adjust SP to make room for 4 words # push $1 # push $2 # push $3 # push $31
	jal Mystery	# in: $2; out: $4
	lw $1, 12($29) **lw $2, 8($29)** **lw $3, 4($29)** **lw $31, 0($29)** **addi $29, $29, 16**	# pop $1 # pop $2 # pop $3 # pop $31 # adjust SP to free up 4 words
	sub $5, $0, $3	#L_A
	addi $5, $5, -1	#L_B
	and $6, $3, $4	#L_C
	and $7, $2, $5	#L_D
	or $8, $6, $7	#L_E
	sw $8, C($1)	#L_F
	bne $1, $0, Loop	
	jr $31	# return to caller

Part B How many words of static memory are read by `Chroma` and how many are written?

# words read: **2 * 64 = 128**	# words written: **64**

Part C Rewrite the instructions at lines L_A through L_F to optimize register usage.

	and $4, $3, $4	# L_C reuse $4 instead of $6
	sub $3, $0, $3	# L_A reuse $3 instead of $5
	addi $3, $3, -1	# L_B reuse $3 instead of $5
	and $2, $2, $3	# L_D reuse $2 instead of $7
	or $2, $2, $4	# L_E reuse $2 instead of $8
	sw $2, C($1)	# L_F saved 4 registers