Problem ST-7 (5 parts) **Reverse Engineering**

Consider the following MIPS code.

label	instruction	comment
	addi $1, $0, 16	# x
	addi $2, $0, 3	# y
	addi $3, $0, 4	# z
	addi $4, $0, 0	# sum
Loop:	addi $1, $1, -1	
	sll $5, $3, 7	
	sll $6, $2, 4	
	add $7, $5, $6	
	add $8, $7, $1	
	sll $8, $8, 2	
	lw $7, Array($8)	
	add $4, $4, $7	
	bne $1, $0, Loop	
	jr $31	

Part A What type of loop is this (e.g., for, while, do while)?

Part B How many iterations does the loop perform?

Part C The loop is accessing a multi-dimensional array. How many columns (Lx) and rows (Ly) does it have? At has at least how many planes (Lz)?

Lx = **Ly =** **Lz ≥**

Part D In terms of the multi-dimensional array, what is being summed up and placed in register $4 by the program?

Part E How can this program be optimized to reduce the total number of instructions executed by 45 instructions (from 149 to 104 instructions)?