

Problem 2

Assembly Language Programming

In this problem, you will write a procedure that computes the average magnitude of the elements of an 8-element array. (This is a function commonly used in image processing applications.) Your procedure will find the absolute value of each element (i.e., its magnitude), sum the magnitudes together, and then divide the result by 8. The array begins at base address 400 and each element of the array is one word long. *Use only the registers described in the table below.* You should not require any registers in addition to those listed. Do not include assembler directives. Your answer should fit in the boxes provided. Use additional space only if necessary. Provide comments.

register	description	register	description
\$1	address of current element	\$4	value of current element of array
\$2	running sum	\$5	result (average magnitude)
\$3	branch predicate register	\$31	return address

Part 2.A To begin, write lines of code that initialize the element address register (\$1) to the address of the first element of the array and that initialize \$2 to 0.

label	instruction	comment
avg-mag:	addi \$1, \$0, 400	# start with address 400
	addi \$2, \$0, 0	# initialize running sum

Part 2.B Next, take the current element, find its magnitude by computing its absolute value (i.e., if it is negative, negate it). Then add the magnitude to the running sum in \$2.

label	instruction	comment
loop:	lw \$4, 0(\$1)	# get next element
	slt \$3, \$4, \$0	# is it < 0?
	beq \$3, \$0, skip1	# if not, skip next inst.
	sub \$4, \$0, \$4	# otherwise, negate it
skip1:	add \$2, \$2, \$4	# add magnitude to sum

Part 2.C Finally, update the element address register (\$1) to the address of the next element of the array. If the last element of the array has not been processed, then loop back to continue summing elements. Otherwise, divide the sum by 8, put the result into register \$5, and exit the procedure by returning to the caller.

label	instruction	comment
	addi \$1, \$1, 4	# update address
	slti \$3, \$1, 432	# still below address
	bne \$3, \$0, loop	# if so, loop
	srl \$5, \$2, 3	# divide sum by 8
	jr \$31	# return to caller