Problem 1 Assembly Language Programming

In this problem, you will write a procedure that computes the maximum value of a 10 element array. The array begins at base address 200 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. Be sure to provide comments.

register	description	register	description
$1	address of current element	$4	value of current element of array
$2	max value found so far	$5	result (maximum value found)
$3	branch predicate register	$31	return address

Part 1.1 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 the first element of the array.

label	instruction	comment
`max:`	`addi $1,$0,200`	`# start with address 200`
	`lw $2,0($1)`	`# $2 gets first element`

Part 1.B Write a code fragment that updates the element address register ($1) to the address of the next element of the array. Then access the value of the current element and replace the maximum value found so far ($2) if necessary.

label	instruction	comment
`loop:`	`addi $1,$1,4`	`# update address`
	`lw $4,0($1)`	`# get next element`
	`slt $3,$2,$4`	`# is max < current elt?`
	`beq $3,$0,skip1`	`# if not, skip next line`
	`add $2,$0,$4`	`# replace max w/ current`

Part 1.C Finally, write a code fragment, such that if the last element of the array has not been processed, then loop back to continue looking for a maximum value. Otherwise, put the maximum value found into register $5 and exit the procedure by returning to the caller.

label	instruction	comment
`skip1:`	`slti $3,$1,236`	`# reached addr limit, yet?`
	`bne $3,$0,loop`	`# if not, loop`
	`add $5,$0,$2`	`# put result in $5`
	`jr $31`	`# return to caller`