MIPS versus VAX					
Saving registers					
	sort:	addi $29,$29, -36	sort:	.word ^m<r2,r3,r4,r5,r6,r7>	
		sw $15, 0($29)			
		sw $16, 4($29)			
		sw $17, 8($29)			
		sw $18,12($29)			
		sw $19,16($29)			
		sw $20,20($29)			
		sw $24,24($29)			
		sw $25,28($29)			
		sw $31,32($29)			
Procedure body					
Move parameters		move $18, $4		moval r7,8(ap)	
		move $20, $5		moval r5,4(ap)	
Outer loop		add $19, $0, $0		clrl r6	
	for1tst:	slt $8, $19, $20	for1tst:	cmpl r6,(r7)	
		beq $8, $0, exit1		bgeq exit1	
Inner loop		addi $17, $19, -1		subl3 r4,r6,#1	
	for2tst:	slti $8, $17, 0	for2tst:		
		bne $8, $0, exit2		blss exit2	
		muli $15, $17, 4		movl r3,(r5)	
		add $16, $18, $15			
		lw $24, 0($16)			
		lw $25, 4($16)		addl3 r2,r4,#1	
		slt $8, $25, $24		cmpl (r3)[r4],(r3)[r2]	
		beq $8, $0, exit2		bleq exit2	
Pass parameters and call		move $4, $18		pushl (r5)	
		move $5, $17		pushl r4	
		jal swap		calls #2,swap	
Inner loop		addi $17, $17, -1		decl r4	
		j for2tst		brb for2tst	
Outer loop	exit2:	addi $19, $19, 1	exit2:	incl r6	
		j for1tst		brb for1tst	
Restoring registers					
	exit1:	lw $15,0($29)			
		lw $16, 4($29)			
		lw $17, 8($29)			
		lw $18,12($29)			
		lw $19,16($29)			
		lw $20,20($29)			
		lw $24,24($29)			
		lw $25,28($29)			
		lw $31,32($29)			
		addi $29,$29, 36			
Procedure return					
		jr $31	exit1:	ret	

FIGURE E.56 MIPS32 versus VAX assembly version of procedure sort in Figure E.55 on page E-33.