

# Lab Report

**ECPE 170 – Computer Systems and Networks – Spring 2017**

**Name:** STEVE GUERRERO

**Lab Topic:** MIPS Assembly Programming (Basic) (LAB #: 010)

- (1) Take two screenshots of the MIPS register panel: one before your program runs, and one after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Before: part1 Program

FP Regs		nt Regs [10]	
Int Regs [10]			
PC	=	0	
EPC	=	0	
Cause	=	0	
BadVAddr	=	0	
Status	=	805371664	
HI	=	0	
LO	=	0	
R0	[r0]	=	0
R1	[at]	=	0
R2	[v0]	=	0
R3	[v1]	=	0
R4	[a0]	=	1
R5	[a1]	=	2147479980
R6	[a2]	=	2147479988
R7	[a3]	=	0
R8	[t0]	=	0
R9	[t1]	=	0
R10	[t2]	=	0
R11	[t3]	=	0
R12	[t4]	=	0
R13	[t5]	=	0
R14	[t6]	=	0
R15	[t7]	=	0
R16	[s0]	=	0
R17	[s1]	=	0
R18	[s2]	=	0
R19	[s3]	=	0
R20	[s4]	=	0
R21	[s5]	=	0
R22	[s6]	=	0
R23	[s7]	=	0
R24	[t8]	=	0
R25	[t9]	=	0
R26	[k0]	=	0
R27	[k1]	=	0

After: part1 Program

FP Regs	nt Regs [10]	
Int Regs [10]		
PC	=	4194436
EPC	=	0
Cause	=	0
BadVAddr	=	0
Status	=	805371664
HI	=	0
LO	=	0
R0	[r0]	= 0
R1	[at]	= 268500992
R2	[v0]	= 10
R3	[v1]	= 0
R4	[a0]	= 1
R5	[a1]	= 2147479980
R6	[a2]	= 2147479988
R7	[a3]	= 0
R8	[t0]	= 0
R9	[t1]	= 15
R10	[t2]	= 10
R11	[t3]	= 7
R12	[t4]	= 2
R13	[t5]	= 18
R14	[t6]	= -3
R15	[t7]	= 25
R16	[s0]	= 0
R17	[s1]	= 37
R18	[s2]	= 5
R19	[s3]	= 15
R20	[s4]	= 8
R21	[s5]	= 30
R22	[s6]	= 45
R23	[s7]	= 0
R24	[t8]	= 0
R25	[t9]	= 0
R26	[k0]	= 0
R27	[k1]	= 0

- (2) Take two screenshots of the MIPS memory panel (data tab): one before your program runs, and one after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. In the after-execution capture, circle the memory location (not register) that contains the final calculated value of Z.

## Before: part1 Program

User data segment [10000000]..[10040000]  
[10000000]..[1003ffff] 00000000

### User Stack [7ffff1a8]..[80000000]

[7ffff1a8]	0000000001	2147480239			. . . . .
[7ffff1b0]	0000000000	2147483637	2147483619	2147483597	. . . . .
[7ffff1c0]	2147483544	2147483490	2147483474	2147483454	. . . . b . . . . R . . . . > . . . .
[7ffff1d0]	2147483437	2147483402	2147483384	2147483316	- . . . . . . . . . . . . . . . .
[7ffff1e0]	2147483293	2147483248	2147483233	2147481817	. . . . . p . . . . a . . . . . . . . . .
[7ffff1f0]	2147481798	2147481740	2147481688	2147481647	. . . . . X . . . . / . . . .
[7ffff200]	2147481596	2147481528	2147481377	2147481354	. . . . . ! . . . . . . . . . .
[7ffff210]	2147481336	2147481303	2147481282	2147481226	. . . . . . . . . . . . . . . .
[7ffff220]	2147481208	2147481188	2147481169	2147481152	x . . . . d . . . . Q . . . . @ . . . .
[7ffff230]	2147481137	2147481083	2147481065	2147481036	l . . . . . . . . . . . . . . . .
[7ffff240]	2147481018	2147480992	2147480961	2147480940	. . . . . . . . . . . . . . l . . . .
[7ffff250]	2147480925	2147480917	2147480902	2147480858	] . . . . U . . . . F . . . . . . . . . .
[7ffff260]	2147480840	2147480808	2147480781	2147480763	. . . . . . . . . . . . . . . .
[7ffff270]	2147480745	2147480643	2147480583	2147480551	. . . . . C . . . . . . . . . . . . . . .
[7ffff280]	2147480536	2147480502	2147480471	2147480460	. . . . . . . . . . . . . . . .
[7ffff290]	2147480434	2147480415	2147480381	2147480342	r . . . . _ . . . . = . . . . . . . . . .
[7ffff2a0]	2147480319	2147480301	0000000000	0788529152	. . . . . . . . . . . . . . . /
[7ffff2b0]	1701670760	1919513391	1986360435	1647275621	h o m e / s i r s t e v e n / b
[7ffff2c0]	1969386601	1952803683	0825242159	1886609207	i t b u c k e t / 2 0 1 7 _ s p
[7ffff2d0]	1735289202	1885562207	0808923493	1650551855	r i n g _ e c p e l 7 0 / l a b
[7ffff2e0]	1882140721	0846492257	1836278062	0792551168	l 0 / p a r t 2 . a s m . _ = /
[7ffff2f0]	0796029813	0795765090	1886614641	1325428073	u s r / b i n / q t s p i m . O
[7ffff300]	1464878156	1747926340	0795176303	1936877939	L D P W D = / h o m e / s i r s
[7ffff310]	1702258036	1096286318	1330140245	1498696018	t e v e n . X A U T H O R I T Y
[7ffff320]	1869098813	1932485997	1953722985	1852143205	= / h o m e / s i r s t e v e n
[7ffff330]	1633168943	1869116533	2037672306	1397050368	/ . X a u t h o r i t y . L E S
[7ffff340]	1330398035	0792544595	0796029813	0795765090	S C L O S E = / u s r / b i n /
[7ffff350]	1936942444	1701865840	0544417056	1191211813	l e s s p i p e % s % s . G
[7ffff360]	1230981972	1330470733	1162630468	1969383741	T K _ I M _ M O D U L E = i b u
[7ffff370]	1146617971	1430478663	1313165906	1162108756	s . X D G _ C U R R E N T _ D E
[7ffff380]	1330924371	1851080016	0007959657	1347635524	S K T O P = U n i t y . D I S P
[7ffff390]	1022505506	1474807354	1281075876	1220261245	t a v - . . . . v r c n t m t r

## After: part1 Program

Data	Text
Data	
User data segment [10000000]..[10040000]	
[10000000]..[1000ffff] 00000000	
[10010000] 0000000015 0000000010	0000000007 0000000002 . . . . .
[10010010] 0000000018 -3 0000000037	0000000000 . . . . . % . . . . .
[10010020]..[1003ffff] 00000000	
User Stack [7ffff1a8]..[80000000]	
[7ffff1a8]	0000000001 2147480239 . . . . .
[7ffff1b0]	0000000000 2147483637 2147483619 2147483597 . . . . .
[7ffff1c0]	2147483544 2147483490 2147483474 2147483454 . . . . b . . . . R . . . . > . . . .
[7ffff1d0]	2147483437 2147483402 2147483384 2147483316 - . . . . . . . . . . . . . . . .
[7ffff1e0]	2147483293 2147483248 2147483233 2147481817 . . . . . p . . . . a . . . . . . . . . .
[7ffff1f0]	2147481798 2147481740 2147481688 2147481647 . . . . . X . . . . / . . . .
[7ffff200]	2147481596 2147481528 2147481377 2147481354 . . . . . ! . . . . . . . . . .
[7ffff210]	2147481336 2147481303 2147481282 2147481226 . . . . . . . . . . . . . . . .
[7ffff220]	2147481208 2147481188 2147481169 2147481152 x . . . . d . . . . Q . . . . @ . . . .
[7ffff230]	2147481137 2147481083 2147481065 2147481036 l . . . . . . . . . . . . . . . .
[7ffff240]	2147481018 2147480992 2147480961 2147480940 . . . . . . . . . . . . . . l . . . .
[7ffff250]	2147480925 2147480917 2147480902 2147480858 ] . . . . U . . . . F . . . . . . . . . .
[7ffff260]	2147480840 2147480808 2147480781 2147480763 . . . . . . . . . . . . . . . .
[7ffff270]	2147480745 2147480643 2147480583 2147480551 . . . . . C . . . . . . . . . . . . . . .
[7ffff280]	2147480536 2147480502 2147480471 2147480460 . . . . . . . . . . . . . . . .
[7ffff290]	2147480434 2147480415 2147480381 2147480342 r . . . . _ . . . . = . . . . . . . . . .
[7ffff2a0]	2147480319 2147480301 0000000000 0795830319 . . . . . . . . . . . . . . / h o /
[7ffff2b0]	1701670760 1919513391 1986360435 1647275621 h o m e / s i r s t e v e n / b
[7ffff2c0]	1969386601 1952803683 0825242159 1886609207 i t b u c k e t / 2 0 1 7 _ s p
[7ffff2d0]	1735289202 1885562207 0808923493 1650551855 r i n g _ e c p e l 7 0 / l a b
[7ffff2e0]	1882140721 0829715041 1836278062 0792551168 l 0 / p a r t 1 . a s m . _ = /
[7ffff2f0]	0796029813 0795765090 1886614641 1325428073 u s r / b i n / q t s p i m . O
[7ffff300]	1464878156 1747926340 0795176303 1936877939 L D P W D = / h o m e / s i r s
[7ffff310]	1702258036 1096286318 1330140245 1498696018 t e v e n . X A U T H O R I T Y
[7ffff320]	1869098813 1932485997 1953722985 1852143205 = / h o m e / s i r s t e v e n
[7ffff330]	1633168943 1869116533 2037672306 1397050368 / . X a u t h o r i t y . L E S
[7ffff340]	1330398035 0792544595 0796029813 0795765090 S C L O S E = / u s r / b i n /
[7ffff350]	1936942444 1701865840 0544417056 1191211813 l e s s p i p e % s % s . G
[7ffff360]	1230981972 1330470733 1162630468 1969383741 T K _ I M _ M O D U L E = i b u

(3) Take two screenshots of the MIPS register panel: one before your program runs, and one after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

Before: part 2 program

FP Regs		nt Regs [10]	
Int Regs [10]			
PC	=	0	
EPC	=	0	
Cause	=	0	
BadVAddr	=	0	
Status	=	805371664	
HI	=	0	
LO	=	0	
R0	[r0]	=	0
R1	[at]	=	0
R2	[v0]	=	0
R3	[v1]	=	0
R4	[a0]	=	1
R5	[a1]	=	2147479980
R6	[a2]	=	2147479988
R7	[a3]	=	0
R8	[t0]	=	0
R9	[t1]	=	0
R10	[t2]	=	0
R11	[t3]	=	0
R12	[t4]	=	0
R13	[t5]	=	0
R14	[t6]	=	0
R15	[t7]	=	0
R16	[s0]	=	0
R17	[s1]	=	0
R18	[s2]	=	0
R19	[s3]	=	0
R20	[s4]	=	0
R21	[s5]	=	0
R22	[s6]	=	0
R23	[s7]	=	0
R24	[t8]	=	0
R25	[t9]	=	0
R26	[k0]	=	0
R27	[k1]	=	0

After: part 2 program

FP Regs		nt Regs [10]	
Int Regs [10]			
PC	=	4194500	
EPC	=	0	
Cause	=	0	
BadVAddr	=	0	
Status	=	805371664	
HI	=	0	
LO	=	0	
R0	[r0]	=	0
R1	[at]	=	268500992
R2	[v0]	=	10
R3	[v1]	=	0
R4	[a0]	=	1
R5	[a1]	=	2147479980
R6	[a2]	=	2147479988
R7	[a3]	=	0
R8	[t0]	=	0
R9	[t1]	=	10
R10	[t2]	=	15
R11	[t3]	=	4
R12	[t4]	=	-1
R13	[t5]	=	0
R14	[t6]	=	0
R15	[t7]	=	0
R16	[s0]	=	0
R17	[s1]	=	0
R18	[s2]	=	0
R19	[s3]	=	0
R20	[s4]	=	0
R21	[s5]	=	0
R22	[s6]	=	0
R23	[s7]	=	0
R24	[t8]	=	0
R25	[t9]	=	0
R26	[k0]	=	0
R27	[k1]	=	0

(4) Take two screenshots of the MIPS memory panel (data tab): one before your program runs, and one after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. In the after-execution capture, circle the memory location (not register) that contains the final calculated value of Z.

Before: part 2 program

**User data segment [10000000]..[10040000]**  
[10000000]..[1003ffff] 00000000

**User Stack [7ffff1a8]..[80000000]**

[7ffff1a8]	0000000001	2147480239				. . . . .
[7ffff1b0]	0000000000	2147483637	2147483619	2147483597		. . . . .
[7ffff1c0]	2147483544	2147483490	2147483474	2147483454		. . . . b . . . R . . . > . . .
[7ffff1d0]	2147483437	2147483402	2147483384	2147483316		- . . . .
[7ffff1e0]	2147483293	2147483248	2147483233	2147481817		. . . . p . . . a . . . .
[7ffff1f0]	2147481798	2147481740	2147481688	2147481647		. . . . . X . . . / . . .
[7ffff200]	2147481596	2147481528	2147481377	2147481354		. . . . . ! . . . .
[7ffff210]	2147481336	2147481303	2147481282	2147481226		. . . . .
[7ffff220]	2147481208	2147481188	2147481169	2147481152		x . . . d . . . Q . . . @ . . .
[7ffff230]	2147481137	2147481083	2147481065	2147481036		l . . . . .
[7ffff240]	2147481018	2147480992	2147480961	2147480940		. . . . . l . . .
[7ffff250]	2147480925	2147480917	2147480902	2147480858		] . . . U . . . F . . . .
[7ffff260]	2147480840	2147480808	2147480781	2147480763		. . . . .
[7ffff270]	2147480745	2147480643	2147480583	2147480551		. . . . C . . . .
[7ffff280]	2147480536	2147480502	2147480471	2147480460		. . . . .
[7ffff290]	2147480434	2147480415	2147480381	2147480342		r . . . _ . . . = . . . .
[7ffff2a0]	2147480319	2147480301	0000000000	0788529152		. . . . . /
[7ffff2b0]	1701670760	1919513391	1986360435	1647275621		h o m e / s i r s t e v e n / b
[7ffff2c0]	1969386601	1952803683	0825242159	1886609207		i t b u c k e t / 2 0 1 7 _ s p
[7ffff2d0]	1735289202	1885562207	0808923493	1650551855		r i n g _ e c p e 1 7 0 / l a b
[7ffff2e0]	1882140721	0846492257	1836278062	0792551168		l 0 / p a r t 2 . a s m . _ = /
[7ffff2f0]	0796029813	0795765090	1886614641	1325428073		u s r / b i n / q t s p i m . O
[7ffff300]	1464878156	1747926340	0795176303	1936877939		L D P W D = / h o m e / s i r s
[7ffff310]	1702258036	1096286318	1330140245	1498696018		t e v e n . X A U T H O R I T Y
[7ffff320]	1869098813	1932485997	1953722985	1852143205		= / h o m e / s i r s t e v e n
[7ffff330]	1633168943	1869116533	2037672306	1397050368		/ . X a u t h o r i t y . L E S
[7ffff340]	1330398035	0792544595	0796029813	0795765090		S C L O S E = / u s r / b i n /
[7ffff350]	1936942444	1701865840	0544417056	1191211813		l e s s p i p e % s % s . G
[7ffff360]	1230981972	1330470733	1162630468	1969383741		T K _ I M _ M O D U L E = i b u
[7ffff370]	1146617971	1430478663	1313165906	1162108756		s . X D G _ C U R R E N T _ D E
[7ffff380]	1330924371	1851080016	0007959657	1347635524		S K T O P = U n i t y . D I S P
[7ffff390]	1028250506	1476407354	1381075876	1320261845		T A X _ . . . V D C _ D U M T T

After: part 2 program

Data	Text
Data	
User data segment [10000000]..[10040000]	
[10000000]..[1000ffff]	00000000
[10010000]	0000000010 0000000015 0000000006 -2 . . . . .
[10010010]..[1003ffff]	00000000
User Stack [7ffff1a8]..[80000000]	
[7ffff1a8]	0000000001 2147480239 . . . . .
[7ffff1b0]	0000000000 2147483637 2147483619 2147483597 . . . . .
[7ffff1c0]	2147483544 2147483490 2147483474 2147483454 . . . . b . . . R . . . > . . .
[7ffff1d0]	2147483437 2147483402 2147483384 2147483316 - . . . . .
[7ffff1e0]	2147483293 2147483248 2147483233 2147481817 . . . . p . . . a . . . . .
[7ffff1f0]	2147481798 2147481740 2147481688 2147481647 . . . . . X . . . / . . .
[7ffff200]	2147481596 2147481528 2147481377 2147481354 . . . . . ! . . . . .
[7ffff210]	2147481336 2147481303 2147481282 2147481226 . . . . .
[7ffff220]	2147481208 2147481188 2147481169 2147481152 x . . . d . . . Q . . . @ . . .
[7ffff230]	2147481137 2147481083 2147481065 2147481036 l . . . . .
[7ffff240]	2147481018 2147480992 2147480961 2147480940 . . . . . l . . .
[7ffff250]	2147480925 2147480917 2147480902 2147480858 j . . . U . . . F . . . . .
[7ffff260]	2147480840 2147480808 2147480781 2147480763 . . . . .
[7ffff270]	2147480745 2147480643 2147480583 2147480551 . . . . C . . . . .
[7ffff280]	2147480536 2147480502 2147480471 2147480460 . . . . .
[7ffff290]	2147480434 2147480415 2147480381 2147480342 r . . . _ . . . = . . . . .
[7ffff2a0]	2147480319 2147480301 0000000000 0788529152 . . . . . /
[7ffff2b0]	1701670760 1919513391 1986360435 1647275621 h o m e / s i r s t e v e n / b
[7ffff2c0]	1969386601 1952803683 0825242159 1886609207 i t b u c k e t / 2 0 1 7 _ s p
[7ffff2d0]	1735289202 1885562207 0808923493 1650551855 r i n g _ e c p e l 7 0 / l a b
[7ffff2e0]	1882140721 0846492257 1836278062 0792551168 l 0 / p a r t 2 . a s m . _ = /
[7ffff2f0]	0796029813 0795765090 1886614641 1325428073 u s r / b i n / q t s p i m . O
[7ffff300]	1464878156 1747926340 0795176303 1936877939 L D P W D = / h o m e / s i r s
[7ffff310]	1702258036 1096286318 1330140245 1498696018 t e v e n . X A U T H O R I T Y
[7ffff320]	1869098813 1932485997 1953722985 1852143205 = / h o m e / s i r s t e v e n
[7ffff330]	1633168943 1869116533 2037672306 1397050368 / . X a u t h o r i t y . L E S
[7ffff340]	1330398035 0792544595 0796029813 0795765090 S C L O S E = / u s r / b i n /
[7ffff350]	1936942444 1701865840 0544417056 1191211813 l e s s p i p e % s % s . G
[7ffff360]	1230981972 1330470733 1162630468 1969383741 T K _ I M _ M O D U L E = i b u
[7ffff370]	1146617971 1430478663 1313165906 1162108756 s . X D G _ C U R R E N T _ D E



(5) Take a screenshot of the MIPS register panel after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.

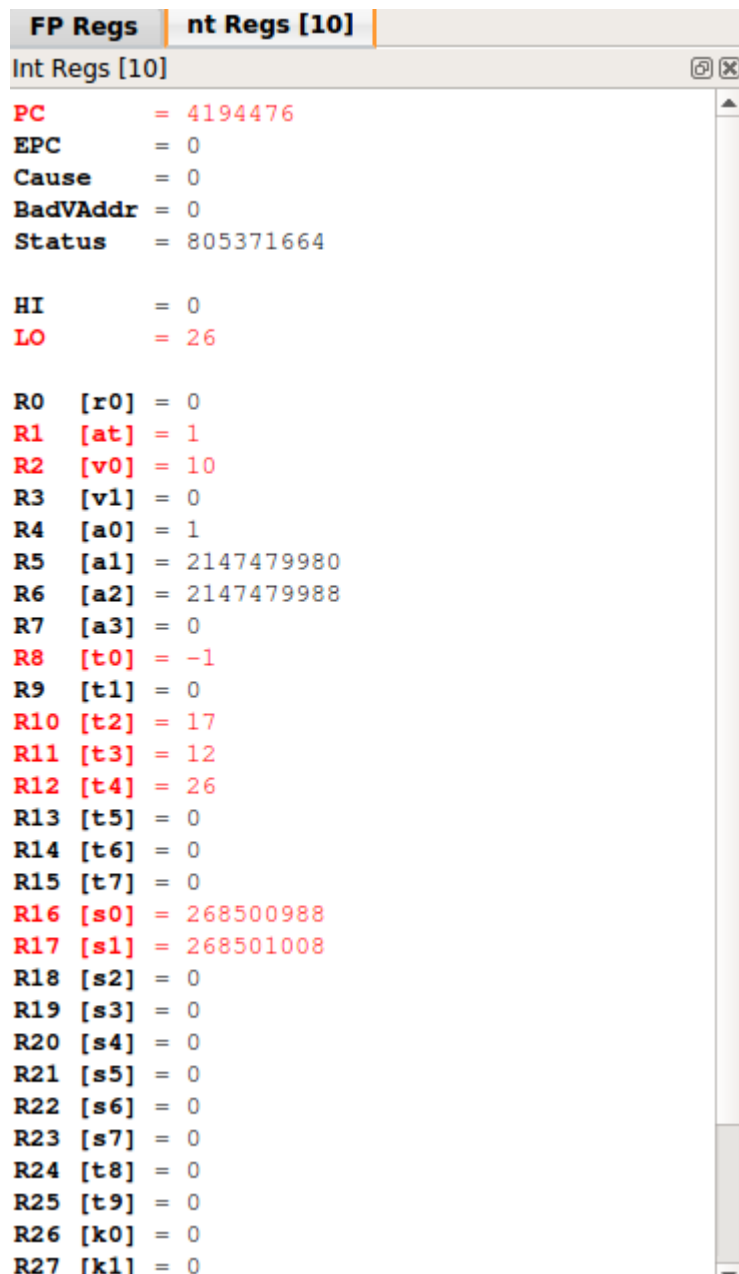
FP Regs		nt Regs [10]	
Int Regs [10]			
PC	=	4194424	
EPC	=	0	
Cause	=	0	
BadVAddr	=	0	
Status	=	805371664	
HI	=	0	
LO	=	0	
R0	[r0]	=	0
R1	[at]	=	268500992
R2	[v0]	=	10
R3	[v1]	=	0
R4	[a0]	=	1
R5	[a1]	=	2147479980
R6	[a2]	=	2147479988
R7	[a3]	=	0
R8	[t0]	=	0
R9	[t1]	=	0
R10	[t2]	=	78
R11	[t3]	=	0
R12	[t4]	=	0
R13	[t5]	=	0
R14	[t6]	=	0
R15	[t7]	=	0
R16	[s0]	=	0
R17	[s1]	=	0
R18	[s2]	=	0
R19	[s3]	=	0
R20	[s4]	=	0
R21	[s5]	=	0
R22	[s6]	=	0
R23	[s7]	=	0
R24	[t8]	=	0
R25	[t9]	=	0
R26	[k0]	=	0
R27	[k1]	=	0

(6) Take a screenshot of the MIPS memory panel (data tab) after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. Circle the memory location (not register) that contains the final calculated values of I and Z.

I is in Green; Z is in Red

Data	Text
Data	
<b>User data segment [10000000]..[10040000]</b>	
[10000000]..[10010003]	00000000
[10010004]	0000000078 0000000000 N . . . . .
[10010010]..[1003ffff]	00000000
<b>User Stack [7ffff1a8]..[80000000]</b>	
[7ffff1a8]	0000000001 2147480239 . . . . .
[7ffff1b0]	0000000000 2147483637 2147483619 2147483597 . . . . .
[7ffff1c0]	2147483544 2147483490 2147483474 2147483454 . . . . b . . . R . . . > . . .
[7ffff1d0]	2147483437 2147483402 2147483384 2147483316 - . . . . .
[7ffff1e0]	2147483293 2147483248 2147483233 2147481817 . . . . p . . . a . . . . .
[7ffff1f0]	2147481798 2147481740 2147481688 2147481647 . . . . . X . . . / . . .
[7ffff200]	2147481596 2147481528 2147481377 2147481354 . . . . . ! . . . . .
[7ffff210]	2147481336 2147481303 2147481282 2147481226 . . . . .
[7ffff220]	2147481208 2147481188 2147481169 2147481152 x . . . d . . . Q . . . @ . . .
[7ffff230]	2147481137 2147481083 2147481065 2147481036 l . . . . .
[7ffff240]	2147481018 2147480992 2147480961 2147480940 . . . . . l . . .
[7ffff250]	2147480925 2147480917 2147480902 2147480858 ] . . . U . . . F . . . . .
[7ffff260]	2147480840 2147480808 2147480781 2147480763 . . . . .
[7ffff270]	2147480745 2147480643 2147480583 2147480551 . . . . C . . . . .
[7ffff280]	2147480536 2147480502 2147480471 2147480460 . . . . .
[7ffff290]	2147480434 2147480415 2147480381 2147480342 r . . . _ . . . = . . . . .
[7ffff2a0]	2147480319 2147480301 0000000000 0788529152 . . . . . /
[7ffff2b0]	1701670760 1919513391 1986360435 1647275621 h o m e / s i r s t e v e n / b
[7ffff2c0]	1969386601 1952803683 0825242159 1886609207 i t b u c k e t / 2 0 1 7 _ s p
[7ffff2d0]	1735289202 1885562207 0808923493 1650551855 r i n g _ e c p e l 7 0 / l a b
[7ffff2e0]	1882140721 0863269473 1836278062 0792551168 l 0 / p a r t 3 . a s m . _ = /
[7ffff2f0]	0796029813 0795765090 1886614641 1325428073 u s r / b i n / q t s p i m . O
[7ffff300]	1464878156 1747926340 0795176303 1936877939 L D P W D = / h o m e / s i r s
[7ffff310]	1702258036 1096286318 1330140245 1498696018 t e v e n . X A U T H O R I T Y
[7ffff320]	1869098813 1932485997 1953722985 1852143205 = / h o m e / s i r s t e v e n
[7ffff330]	1633168943 1869116533 2037672306 1397050368 / . X a u t h o r i t y . L E S
[7ffff340]	1330398035 0792544595 0796029813 0795765090 S C L O S E = / u s r / b i n /
[7ffff350]	1936942444 1701865840 0544417056 1191211813 l e s s p i p e % s % s . G
[7ffff360]	1230981972 1330470733 1162630468 1969383741 T K _ I M _ M O D U L E = i b u
[7ffff370]	1146617971 1430478663 1313165906 1162108756 s . X D G _ C U R R E N T _ D E

(7) Take a screenshot of the MIPS register panel after your program finishes. Put the register panel in Decimal mode (right-click) so it is easy to see register values.



The screenshot shows a window titled "FP Regs" and "nt Regs [10]". Below the title bar is a tab labeled "Int Regs [10]". The main area displays the values of various MIPS registers in decimal format. The registers are listed in two columns. The first column contains the register name and its value, and the second column contains the register name and its value. The registers are listed in the following order: PC, EPC, Cause, BadVAddr, Status, HI, LO, R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23, R24, R25, R26, and R27. The values are as follows:

Register	Value
PC	4194476
EPC	0
Cause	0
BadVAddr	0
Status	805371664
HI	0
LO	26
R0 [r0]	0
R1 [at]	1
R2 [v0]	10
R3 [v1]	0
R4 [a0]	1
R5 [a1]	2147479980
R6 [a2]	2147479988
R7 [a3]	0
R8 [t0]	-1
R9 [t1]	0
R10 [t2]	17
R11 [t3]	12
R12 [t4]	26
R13 [t5]	0
R14 [t6]	0
R15 [t7]	0
R16 [s0]	268500988
R17 [s1]	268501008
R18 [s2]	0
R19 [s3]	0
R20 [s4]	0
R21 [s5]	0
R22 [s6]	0
R23 [s7]	0
R24 [t8]	0
R25 [t9]	0
R26 [k0]	0
R27 [k1]	0

(8) Take a screenshot of the MIPS memory panel (data tab) after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. Circle the final values of array A.

Data	Text			
Data				
User data segment [10000000]..[10040000]				
[10000000]..[1000ffff]	00000000			
[10010000]	0000000026	0000000028	0000000030	0000000032
[10010010]	0000000034	0000000001	0000000002	0000000003
[10010020]	0000000004	0000000005	0000000012	0000000000
[10010030]..[1003ffff]	00000000			
User Stack [7ffff1a8]..[80000000]				
[7ffff1a8]	0000000001	2147480239		
[7ffff1b0]	0000000000	2147483637	2147483619	2147483597
[7ffff1c0]	2147483544	2147483490	2147483474	2147483454
[7ffff1d0]	2147483437	2147483402	2147483384	2147483316
[7ffff1e0]	2147483293	2147483248	2147483233	2147481817
[7ffff1f0]	2147481798	2147481740	2147481688	2147481647
[7ffff200]	2147481596	2147481528	2147481377	2147481354
[7ffff210]	2147481336	2147481303	2147481282	2147481226
[7ffff220]	2147481208	2147481188	2147481169	2147481152
[7ffff230]	2147481137	2147481083	2147481065	2147481036
[7ffff240]	2147481018	2147480992	2147480961	2147480940
[7ffff250]	2147480925	2147480917	2147480902	2147480858
[7ffff260]	2147480840	2147480808	2147480781	2147480763
[7ffff270]	2147480745	2147480643	2147480583	2147480551
[7ffff280]	2147480536	2147480502	2147480471	2147480460
[7ffff290]	2147480434	2147480415	2147480381	2147480342
[7ffff2a0]	2147480319	2147480301	0000000000	0788529152
[7ffff2b0]	1701670760	1919513391	1986360435	1647275621
[7ffff2c0]	1969386601	1952803683	0825242159	1886609207
[7ffff2d0]	1735289202	1885562207	0808923493	1650551855
[7ffff2e0]	1882140721	0880046689	1836278062	0792551168
[7ffff2f0]	0796029813	0795765090	1886614641	1325428073
[7ffff300]	1464878156	1747926340	0795176303	1936877939
[7ffff310]	1702258036	1096286318	1330140245	1498696018
[7ffff320]	1869098813	1932485997	1953722985	1852143205
[7ffff330]	1633168943	1869116533	2037672306	1397050368
[7ffff340]	1330398035	0792544595	0796029813	0795765090
[7ffff350]	1036042444	1701865840	0544417056	1101211812

(9) Take a screenshot of the MIPS memory panel (data tab) after your program finishes. Put the memory panel in Hex mode (right-click), since Decimal mode will not allow us to distinguish between bytes. Circle two things: the final value of the pointer 'result' in memory, and the corresponding location that result points to. Does that location in memory contain the ASCII code for the character 'e'? (If not, you had better check your work!)

ASCII code for 'e' in Hex = 65

Data	Text				
Data					
User data segment [10000000]..[10040000]					
[10000000]..[1000ffff]	00000000				
[10010000]	736e6f43	746e6174	706f6e69	000a656c	C o n s t a n t i n o p l e . .
[10010010]..[10010103]	00000000				
[10010104]	00000065	65746e45	20612072		e . . . E n t e r a
[10010110]	69727473	7520676e	6f742070	36353220	s t r i n g u p t o 2 5 6
[10010120]	61686320	74636172	3a737265	69460020	c h a r a c t e r s : . F i
[10010130]	20747372	6374616d	74612068	64646120	r s t m a t c h a t a d d
[10010140]	73736572	0a00203a	20656854	6374616d	r e s s : . . T h e m a t c
[10010150]	676e6968	61686320	74636172	69207265	h i n g c h a r a c t e r i
[10010160]	00203a73	206f4e0a	6374616d	6f662068	s : . . N o m a t c h f o
[10010170]	20646e75	00000000	00000000	00000000	u n d . . . . .
[10010180]..[1003ffff]	00000000				
User Stack [7ffff1a8]..[80000000]					
[7ffff1a8]	00000001	7ffff2af			. . . . .
[7ffff1b0]	00000000	7fffff5	7fffffe3	7fffffcd	. . . . .
[7ffff1c0]	7fffff98	7fffff62	7fffff52	7fffff3e	. . . . b . . . R . . . > . . .
[7ffff1d0]	7fffff2d	7fffff0a	7ffffef8	7ffffeb4	- . . . .
[7ffff1e0]	7ffffe9d	7ffffe70	7ffffe61	7ffff8d9	. . . . p . . . a . . . .
[7ffff1f0]	7ffff8c6	7ffff88c	7ffff858	7ffff82f	. . . . . X . . . / . . .
[7ffff200]	7ffff7fc	7ffff7b8	7ffff721	7ffff70a	. . . . . ! . . . .
[7ffff210]	7ffff6f8	7ffff6d7	7ffff6c2	7ffff68a	. . . . .
[7ffff220]	7ffff678	7ffff664	7ffff651	7ffff640	x . . . d . . . Q . . . @ . . .
[7ffff230]	7ffff631	7ffff5fb	7ffff5e9	7ffff5cc	l . . . .
[7ffff240]	7ffff5ba	7ffff5a0	7ffff581	7ffff56c	. . . . . l . . .
[7ffff250]	7ffff55d	7ffff555	7ffff546	7ffff51a	] . . . U . . . F . . . .
[7ffff260]	7ffff508	7ffff4e8	7ffff4cd	7ffff4bb	. . . . .
[7ffff270]	7ffff4a9	7ffff443	7ffff407	7ffff3e7	. . . . C . . . .
[7ffff280]	7ffff3d8	7ffff3b6	7ffff397	7ffff38c	. . . . .
[7ffff290]	7ffff372	7ffff35f	7ffff33d	7ffff316	r . . . _ . . . = . . . .
[7ffff2a0]	7ffff2ff	7ffff2ed	00000000	2f000000	. . . . . /
[7ffff2b0]	656d6f68	7269732f	76657473	622f6e65	h o m e / s i r s t e v e n / b
[7ffff2c0]	75627469	74656b63	3130322f	70735f37	i t b u c k e t / 2 0 1 7 _ s p
[7ffff2d0]	676e6972	7063655f	30373165	62616c2f	r i n g _ e c p e l 7 0 / l a b
[7ffff2e0]	702f3031	35747261	6d73612e	2f3d5f00	l 0 / p a r t 5 . a s m . _ = /