

Lab Report

ECPE 170 – Computer Systems and Networks – Spring 2022

Name: Thomas Lau

Lab Topic: MIPS Assembly Programming (Basic) (Lab #: 9)

Question #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.

Answer:

Before:

FP Regs

Int 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] = 2147480392

R6 [a2] = 2147480400

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

R28 [gp] = 268468224

R29 [sp] = 2147480388

R30 [s8] = 0

R31 [ra] = 0

After:

FP Regs

Int Regs [10]

Int Regs [10]

PC = 4194408

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] = 2147480392

R6 [a2] = 2147480400

R7 [a3] = 0

R8 [t0] = 37

R9 [t1] = 7

R10 [t2] = 7

R11 [t3] = 8

R12 [t4] = 15

R13 [t5] = -3

R14 [t6] = 25

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

R28 [gp] = 268468224

R29 [sp] = 2147480388

R30 [s8] = 0

R31 [ra] = 4194328

Question #2:

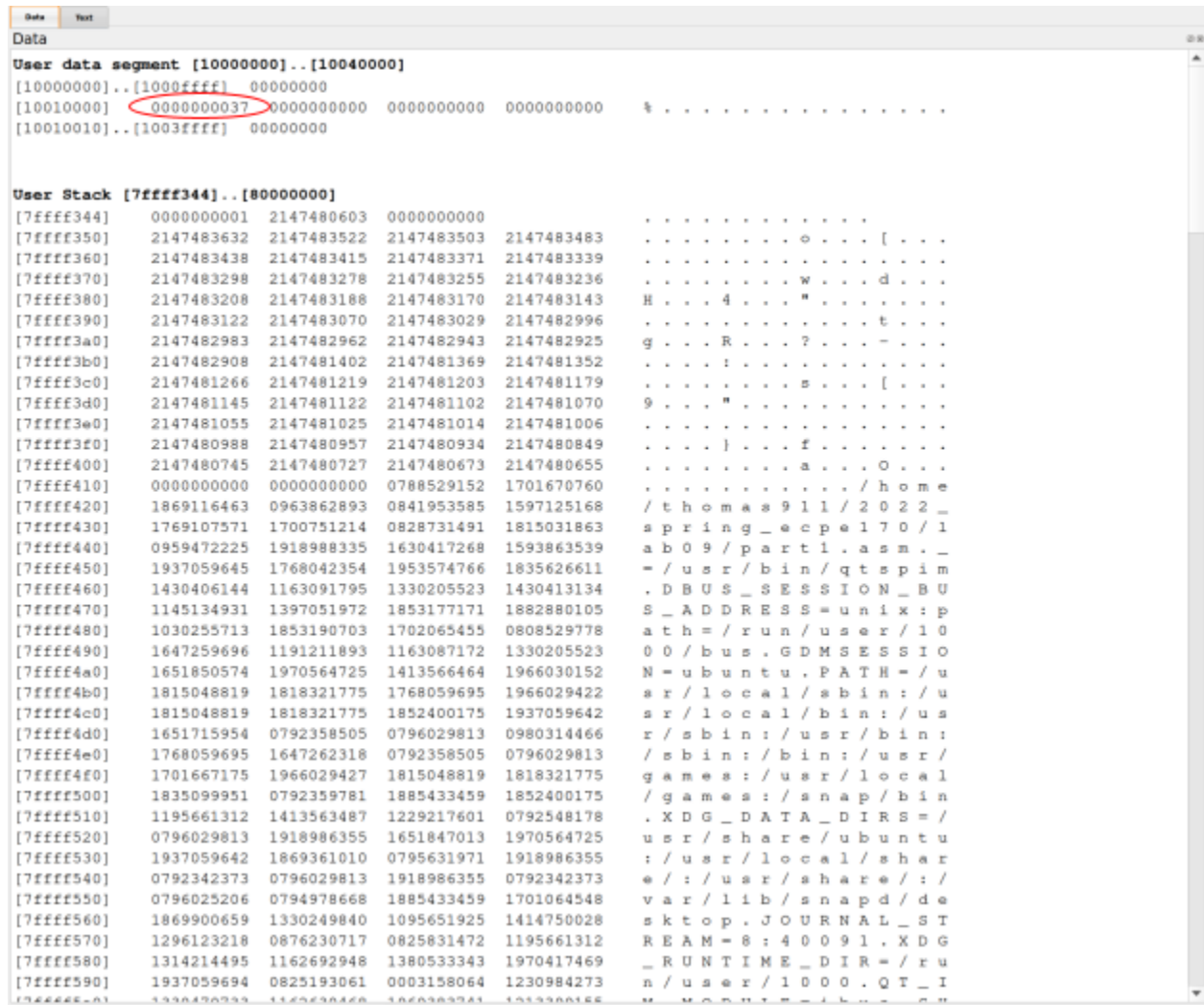
Take two screenshots of the MIPS memory panel (data tab): one before your program runs, and one after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. In the after-execution capture, circle the memory location (not register) that contains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

Answer:

Before:

Data	Text
Data	
User data segment [10000000]..[10040000]	
[10000000]..[1003ffff] 00000000	
User Stack [7ffff344]..[80000000]	
[7ffff344]	0000000001 2147480603 0000000000
[7ffff350]	2147483632 2147483522 2147483503 2147483483 o . . . [. . .
[7ffff360]	2147483438 2147483415 2147483371 2147483339
[7ffff370]	2147483298 2147483278 2147483255 2147483236 w . . . d . . .
[7ffff380]	2147483208 2147483188 2147483170 2147483143 H . . . 4 . . . "
[7ffff390]	2147483122 2147483070 2147483029 2147482996 t . . .
[7ffff3a0]	2147482983 2147482962 2147482943 2147482925 g . . . R . . . ? . . . - . . .
[7ffff3b0]	2147482908 2147481402 2147481369 2147481352 :
[7ffff3c0]	2147481266 2147481219 2147481203 2147481179 s . . . [. . .
[7ffff3d0]	2147481145 2147481122 2147481102 2147481070 9 . . . "
[7ffff3e0]	2147481055 2147481025 2147481014 2147481006
[7ffff3f0]	2147480988 2147480957 2147480934 2147480849 } . . . f
[7ffff400]	2147480745 2147480727 2147480673 2147480655 a . . . O . . .
[7ffff410]	0000000000 0000000000 0788529152 1701670760 / h o m e
[7ffff420]	1869116463 0963862893 0841953585 1597125168 / t h o m a s 9 1 1 / 2 0 2 2 _
[7ffff430]	1769107571 1700751214 0828731491 1815031863 s p r i n g _ e c p e 1 7 0 / l
[7ffff440]	0959472225 1918988335 1630417268 1593863539 a b 0 9 / p a r t 1 . a s m . _
[7ffff450]	1937059645 1768042354 1953574766 1835626611 = / u s r / b i n / q t s p i m
[7ffff460]	1430406144 1163091795 1330205523 1430413134 . D B U S _ S E S S I O N _ B U
[7ffff470]	1145134931 1397051972 1853177171 1882880105 S _ A D D R E S S = u n i x : p
[7ffff480]	1030255713 1853190703 1702065455 0808529778 a t h = / r u n / u s e r / 1 0
[7ffff490]	1647259696 1191211893 1163087172 1330205523 0 0 / b u s . G D M S E S S I O
[7ffff4a0]	1651850574 1970564725 1413566464 1966030152 N = u b u n t u . P A T H = / u
[7ffff4b0]	1815048819 1818321775 1768059695 1966029422 s r / l o c a l / s b i n : / u
[7ffff4c0]	1815048819 1818321775 1852400175 1937059642 s r / l o c a l / b i n : / u s
[7ffff4d0]	1651715954 0792358505 0796029813 0980314466 r / s b i n : / u s r / b i n :
[7ffff4e0]	1768059695 1647262318 0792358505 0796029813 / s b i n : / b i n : / u s r /
[7ffff4f0]	1701667175 1966029427 1815048819 1818321775 g a m e s : / u s r / l o c a l
[7ffff500]	1835099951 0792359781 1885433459 1852400175 / g a m e s : / s n a p / b i n
[7ffff510]	1195661312 1413563487 1229217601 0792548178 . X D G _ D A T A _ D I R S = /
[7ffff520]	0796029813 1918986355 1651847013 1970564725 u s r / s h a r e / u b u n t u
[7ffff530]	1937059642 1869361010 0795631971 1918986355 : / u s r / l o c a l / s h a r
[7ffff540]	0792342373 0796029813 1918986355 0792342373 e / : / u s r / s h a r e / : /
[7ffff550]	0796025206 0794978668 1885433459 1701064548 v a r / l i b / s n a p d / d e
[7ffff560]	1869900659 1330249840 1095651925 1414750028 s k t o p . J O U R N A L _ S T
[7ffff570]	1296123218 0876230717 0825831472 1195661312 R E A M = 8 : 4 0 0 9 1 . X D G
[7ffff580]	1314214495 1162692948 1380533343 1970417469 _ R U N T I M E _ D I R = / r u
[7ffff590]	1937059694 0825193061 0003158064 1230984273 n / u s e r / 1 0 0 0 . Q T _ I
[7ffff5a0]	1330470733 1162630468 1969383741 1213399155 M _ M O D U L E = i b u s . S H

After:



Question #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.

Answer:

Before:

FP Regs

Int 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] = 2147480356

R6 [a2] = 2147480364

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

R28 [gp] = 268468224

R29 [sp] = 2147480352

R30 [s8] = 0

R31 [ra] = 0

After:

FP Regs

Int Regs [10]

Int Regs [10]

PC = 4194476

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] = 2147480356

R6 [a2] = 2147480364

R7 [a3] = 0

R8 [t0] = 10

R9 [t1] = 15

R10 [t2] = 7

R11 [t3] = 0

R12 [t4] = 2

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

R28 [gp] = 268468224

R29 [sp] = 2147480352

R30 [s8] = 0

R31 [ra] = 4194328

Question #4:

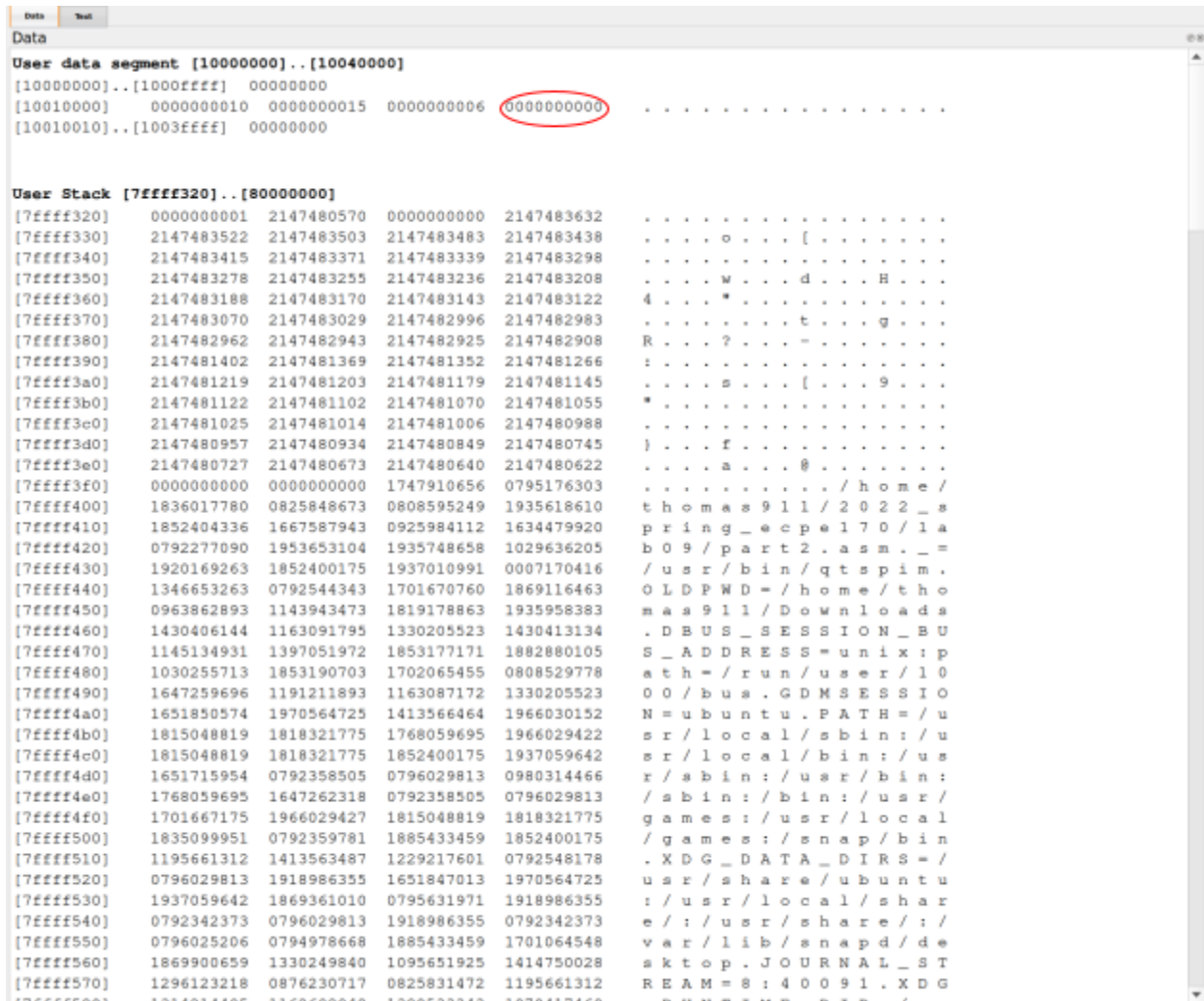
Take two screenshots of the MIPS memory panel (data tab): one before your program runs, and one after your program finishes. Put the memory panel in Decimal mode (right-click), so it is easy to see memory values. In the after-execution capture, circle the memory location (not register) that contains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

Answer:

Before:

Data	Text			
Data				
User data segment [10000000]..[10040000]				
[10000000]..[1000ffff]	00000000			
[10010000]	0000000010	0000000015	0000000006	0000000000
[10010010]..[1003ffff]	00000000			
User Stack [7ffff320]..[80000000]				
[7ffff320]	0000000001	2147480570	0000000000	2147483632
[7ffff330]	2147483522	2147483503	2147483483	2147483438
[7ffff340]	2147483415	2147483371	2147483339	2147483298
[7ffff350]	2147483278	2147483255	2147483236	2147483208
[7ffff360]	2147483188	2147483170	2147483143	2147483122
[7ffff370]	2147483070	2147483029	2147482996	2147482983
[7ffff380]	2147482962	2147482943	2147482925	2147482908
[7ffff390]	2147481402	2147481369	2147481352	2147481266
[7ffff3a0]	2147481219	2147481203	2147481179	2147481145
[7ffff3b0]	2147481122	2147481102	2147481070	2147481055
[7ffff3c0]	2147481025	2147481014	2147481006	2147480988
[7ffff3d0]	2147480957	2147480934	2147480849	2147480745
[7ffff3e0]	2147480727	2147480673	2147480640	2147480622
[7ffff3f0]	0000000000	0000000000	1747910656	0795176303
[7ffff400]	1836017780	0825848673	0808595249	1935618610
[7ffff410]	1852404336	1667587943	0925984112	1634479920
[7ffff420]	0792277090	1953653104	1935748657	1029636205
[7ffff430]	1920169263	1852400175	1937010991	0007170416
[7ffff440]	1346653263	0792544343	1701670760	1869116463
[7ffff450]	0963862893	1143943473	1819178863	1935958383
[7ffff460]	1430406144	1163091795	1330205523	1430413134
[7ffff470]	1145134931	1397051972	1853177171	1882880105
[7ffff480]	1030255713	1853190703	1702065455	0808529778
[7ffff490]	1647259696	1191211893	1163087172	1330205523
[7ffff4a0]	1651850574	1970564725	1413566464	1966030152
[7ffff4b0]	1815048819	1818321775	1768059695	1966029422
[7ffff4c0]	1815048819	1818321775	1852400175	1937059642
[7ffff4d0]	1651715954	0792358505	0796029813	0980314466
[7ffff4e0]	1768059695	1647262318	0792358505	0796029813
[7ffff4f0]	1701667175	1966029427	1815048819	1818321775
[7ffff500]	1835099951	0792359781	1885433459	1852400175
[7ffff510]	1195661312	1413563487	1229217601	0792548178
[7ffff520]	0796029813	1918986355	1651847013	1970564725
[7ffff530]	1937059642	1869361010	0795631971	1918986355
[7ffff540]	0792342373	0796029813	1918986355	0792342373
[7ffff550]	0796025206	0794978668	1885433459	1701064548
[7ffff560]	1869900659	1330249840	1095651925	1414750028
[7ffff570]	1296123218	0876230717	0825831472	1195661312
[7ffff580]	1214214435	1162022448	1380522242	1074417460

After:



The screenshot shows a debugger window with two main sections: 'Data' and 'User Stack'. The 'Data' section displays a memory segment from [10000000] to [10040000]. The first line of data is [10000000]..[1000ffff] 00000000. The second line is [10010000] 0000000010 0000000015 0000000006 0000000000, where the last value '0000000000' is circled in red. The third line is [10010010]..[1003ffff] 00000000. The 'User Stack' section displays a memory segment from [7ffff320] to [80000000]. It shows a list of memory addresses and their corresponding values, which appear to be pointers or addresses. The values are listed in a columnar format, with some values being repeated or truncated. The stack content includes various system paths and identifiers, such as /home/thomas911/2022_spring_ecpe170/lab09/part2.asm, /usr/bin/qtspim, and /usr/local/bin: /usr/bin: /sbin: /bin: /usr/games: /snap/bin. The stack also contains some system variables like XDG_DATA_DIRS and XDG_DATA_DIRS.

Question #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.

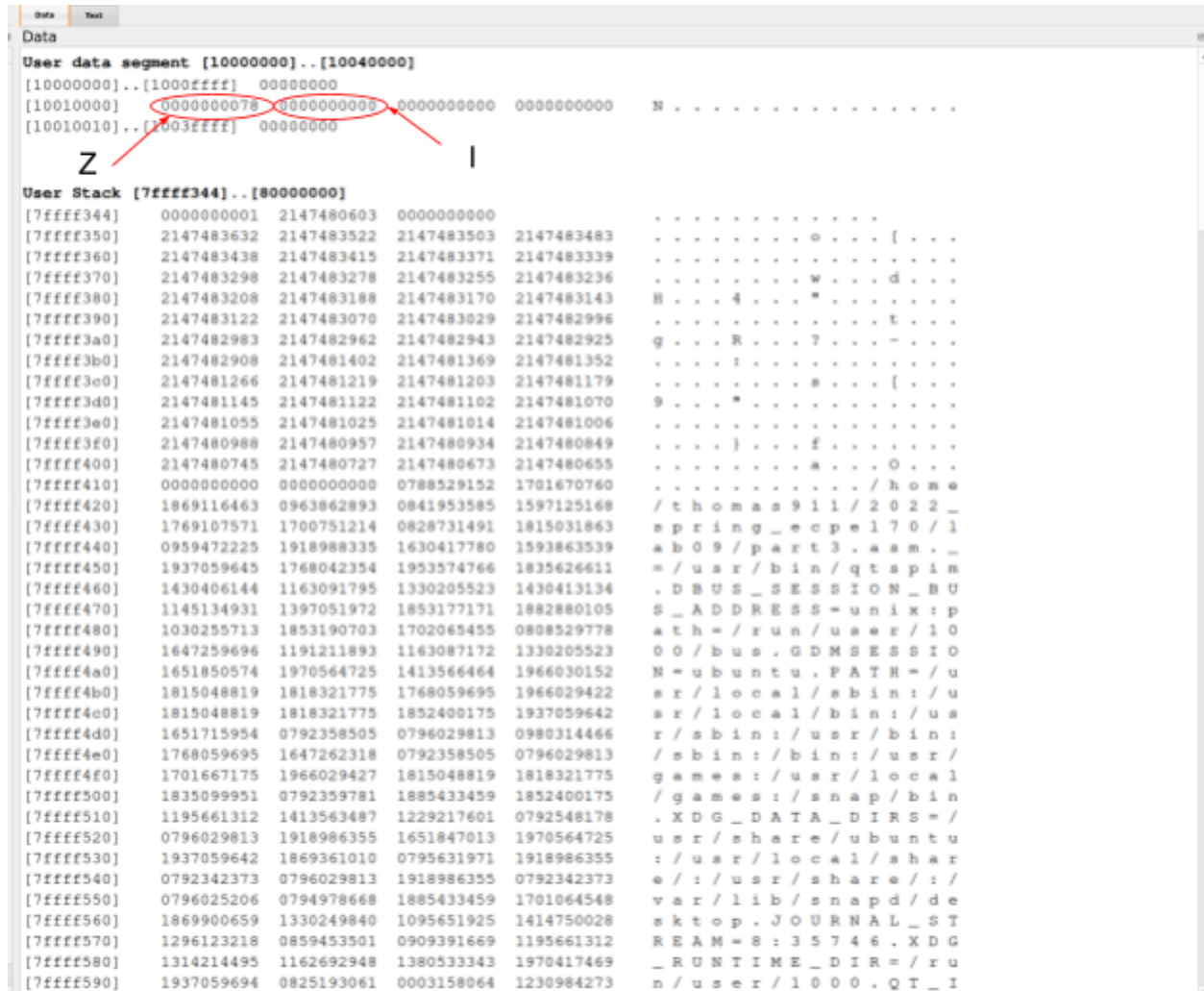
Answer:

Int Regs [10]			
PC		=	4194440
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]	=	2147480392
R6	[a2]	=	2147480400
R7	[a3]	=	0
R8	[t0]	=	78
R9	[t1]	=	0
R10	[t2]	=	20
R11	[t3]	=	100
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
R28	[gp]	=	268468224
R29	[sp]	=	2147480388
R30	[s8]	=	0
R31	[ra]	=	4194328

Question #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.

Answer:



Question #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.

Answer:

FP Regs

Int Regs [10]

Int Regs [10]

PC = 4194456

EPC = 0

Cause = 0

BadVAddr = 0

Status = 805371664

HI = 0

LO = 0

R0 [r0] = 0

R1 [at] = 1

R2 [v0] = 10

R3 [v1] = 0

R4 [a0] = 1

R5 [a1] = 2147480392

R6 [a2] = 2147480400

R7 [a3] = 0

R8 [t0] = 268500988

R9 [t1] = -1

R10 [t2] = 12

R11 [t3] = 268501032

R12 [t4] = 26

R13 [t5] = 17

R14 [t6] = 0

R15 [t7] = 0

R16 [s0] = 268500992

R17 [s1] = 268501012

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

R28 [gp] = 268468224

R29 [sp] = 2147480388

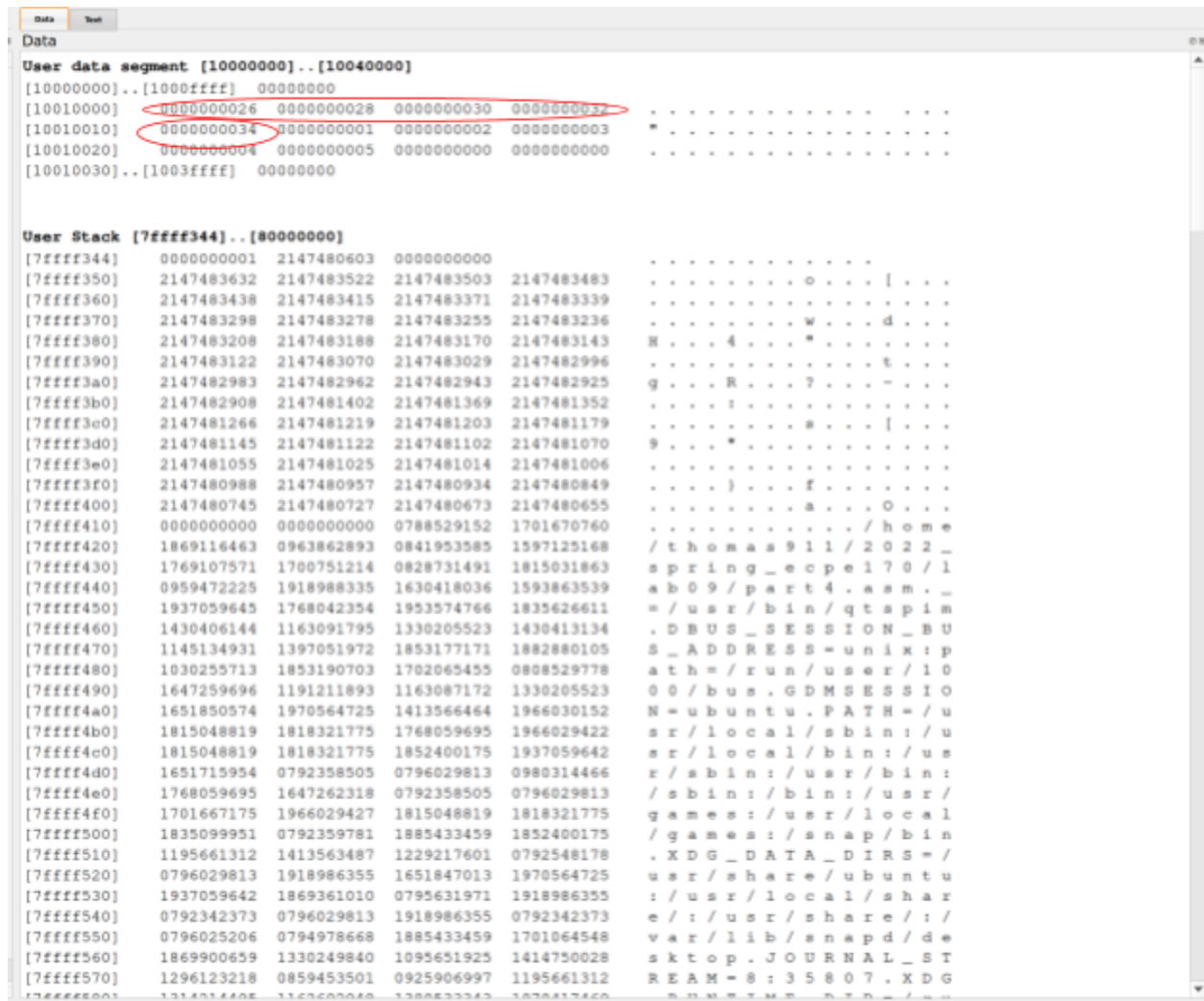
R30 [s8] = 0

R31 [ra] = 4194328

Question #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.

Answer:



Question #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!)

Answer:

Addr	Text
Data	
User data segment [10000000]..[10040000]	
[10000000]..[1000ffff]	00000000
[10010000]	61687465 6664616e 0000000a 00000000 e t h a n a d f
[10010010]..[100100ff]	00000000
[10010100]	6d206f4e 68637461 756f6620 000a646e N o m a t c h f o u n d . .
[10010110]	73726946 616d2074 20686374 6e756f66 F i r s t m a t c h f o u n
[10010120]	74612064 64646120 73736572 540a0020 d a t a d r e s s . . T
[10010130]	6d206568 68637461 20676e69 72616863 h e m a t c h i n g c h a r
[10010140]	65746361 73692072 000a6520 00000000 a c t e r i s e
[10010150]..[1003ffff]	00000000
User Stack [7ffff344]..[80000000]	
[7ffff344]	00000001 7ffff41b 00000000
[7ffff350]	7fffff0 7ffff82 7ffff6f 7ffff5b 0 . . . [. . .
[7ffff360]	7ffff2e 7ffff17 7ffffeb 7ffffcb
[7ffff370]	7ffffa2 7ffff8e 7ffff77 7ffff64 w . . . d . . .
[7ffff380]	7ffff48 7ffff34 7ffff22 7ffff07 R . . . 4 . . . "
[7ffff390]	7ffffd2 7ffffdb 7ffffd95 7ffffd74 t
[7ffff3a0]	7ffffd67 7ffffd52 7ffffd3f 7ffffd2d g . . . R . . . ? . . . - . . .
[7ffff3b0]	7ffffd1c 7ffff73a 7ffff719 7ffff708 i
[7ffff3c0]	7ffff6b2 7ffff683 7ffff673 7ffff65b s [. . .
[7ffff3d0]	7ffff639 7ffff622 7ffff60e 7ffff5ee 9 . . . "
[7ffff3e0]	7ffff5df 7ffff5c1 7ffff5b6 7ffff5ae
[7ffff3f0]	7ffff59c 7ffff57d 7ffff566 7ffff511 } . . . f
[7ffff400]	7ffff4a9 7ffff497 7ffff461 7ffff44f a . . . O . . .
[7ffff410]	00000000 00000000 2f000000 656d6f68 / h o m e
[7ffff420]	6f68742f 3973616d 322f3131 5f323230 / t h o m a s 9 1 1 / 2 0 2 2 _
[7ffff430]	69727073 655f676e 31657063 6c2f3037 s p r i n g _ e c p e l 7 0 / l
[7ffff440]	39306261 7261702f 612e3574 5f006d73 a b 0 9 / p a r t 5 . a s m . _
[7ffff450]	73752f3d 69622f72 74712f6e 6d697073 = / u s r / b i n / q t s p i m
[7ffff460]	55424400 45535f53 4f495353 55425f4e . D B U S _ S E S S I O N _ B U
[7ffff470]	44415f53 53455244 6e753d53 703a7869 S _ A D D R E S S = u n i x : p
[7ffff480]	3d687461 6e75722f 6573752f 30312f72 a t h = / r u n / u s e r / l 0
[7ffff490]	622f3030 47007375 45534d44 4f495353 0 0 / b u s . G D M S E S S I O
[7ffff4a0]	62753d4e 75746e75 54415000 752f3d48 N = u b u n t u . P A T H = / u
[7ffff4b0]	6c2f7273 6c61636f 6962732f 752f3a6e s r / l o c a l / s b i n : / u
[7ffff4c0]	6c2f7273 6c61636f 6e69622f 73752f3a s r / l o c a l / b i n : / u s
[7ffff4d0]	62732f72 2f3a6e69 2f727375 3a6e6962 r / s b i n : / u s r / b i n :
[7ffff4e0]	6962732f 622f3a6e 2f3a6e69 2f727375 / s b i n : / b i n : / u s r /
[7ffff4f0]	656d6167 752f3a73 6c2f7273 6c61636f g a m e s : / u s r / l o c a l
[7ffff500]	6d61672f 2f3a7365 70616e73 6e69622f / g a m e s : / s n a p / b i n
[7ffff510]	47445800 5441445f 49445f41 2f3d5352 . X D G _ D A T A _ D I R S = /
[7ffff520]	2f727375 72616873 62752f65 75746e75 u s r / s h a r e / u b u n t u
[7ffff530]	73752f3a 6f6c2f72 2f6c6163 72616873 : / u s r / l o c a l / s h a r