



# Assessment Module-5

Rana, Karan

CSIT\_230SP21

Instructor: Dr. G.E. Antoniou

TA: Dev Dave

4,4,2021

## Problem 1:

Write a MIPS Assembly program to ADD the numbers: 1, 2, 3, 4, 5, 6 (as in ex-1)

- Use ONLY the instructions: add and li
- The result of the addition (in decimal) should be in register: \$t1

```

1
2 .text
3
4     li $s1, 1      # Assigning 1
5     li $s2, 2      # Assigning 2
6     add $s3,$s1,$s2 # Sum of 1 and 2 storing in $s3
7     li $s4, 3      # Assigning 3
8     li $s5, 4      # Assigning 4
9     add $s6,$s4,$s5 # Sum of 3 and 4 storing in $s6
10    li $t2, 5       # Assigning 5
11    li $t3, 6       # Assigning 6
12    add $t4,$t2,$t3  # Sum of 5 and 6; storing in $t4
13    add $t5,$s3,$s6  # Sum of $s3(line 6) and $s6(line 9); storing the value in $t5
14    add $t1,$t5,$t4  # Sum of $t4(line 12) and $t5(line 13); Storing the value in $t1
15    syscall

```

File

Execute

Text Segment

PC	Address	Code	Basic	Source
0x04000000	0x2410000	addiu \$t1,\$0,1	41	li \$s1, 1 # Assigning 1
0x04000004	0x2410002	addiu \$t1,\$0,2	51	li \$s2, 2 # Assigning 2
0x04000008	0x0232920	add \$t3,\$t1,\$t2	61	add \$s3,\$s1,\$s2 # Sum of 1 and 2 storing in \$s3
0x0400000c	0x2410003	addiu \$t1,\$0,3	71	li \$s4, 3 # Assigning 3
0x04000010	0x2410004	addiu \$t1,\$0,4	81	li \$s5, 4 # Assigning 4
0x04000014	0x0236b20	add \$t6,\$t3,\$t5	91	add \$s6,\$s4,\$s5 # Sum of 3 and 4 storing in \$s6
0x04000018	0x2410005	addiu \$t1,\$0,5	101	li \$t2, 5 # Assigning 5
0x0400001c	0x2410006	addiu \$t1,\$0,6	111	li \$t3, 6 # Assigning 6
0x04000020	0x014b420	add \$t4,\$t2,\$t3	121	add \$t4,\$t2,\$t3 # Sum of 5 and 6; storing in \$t4
0x04000024	0x0236b20	add \$t5,\$t3,\$t6	131	add \$t5,\$s3,\$s6 # Sum of \$s3(line 6) and \$s6(line 9); storing the value in \$t5
0x04000028	0x014b420	add \$t1,\$t5,\$t4	141	add \$t1,\$t5,\$t4 # Sum of \$t4(line 12) and \$t5(line 13); Storing the value in \$t1
0x0400002c	0x000000c	syscall	151	syscall

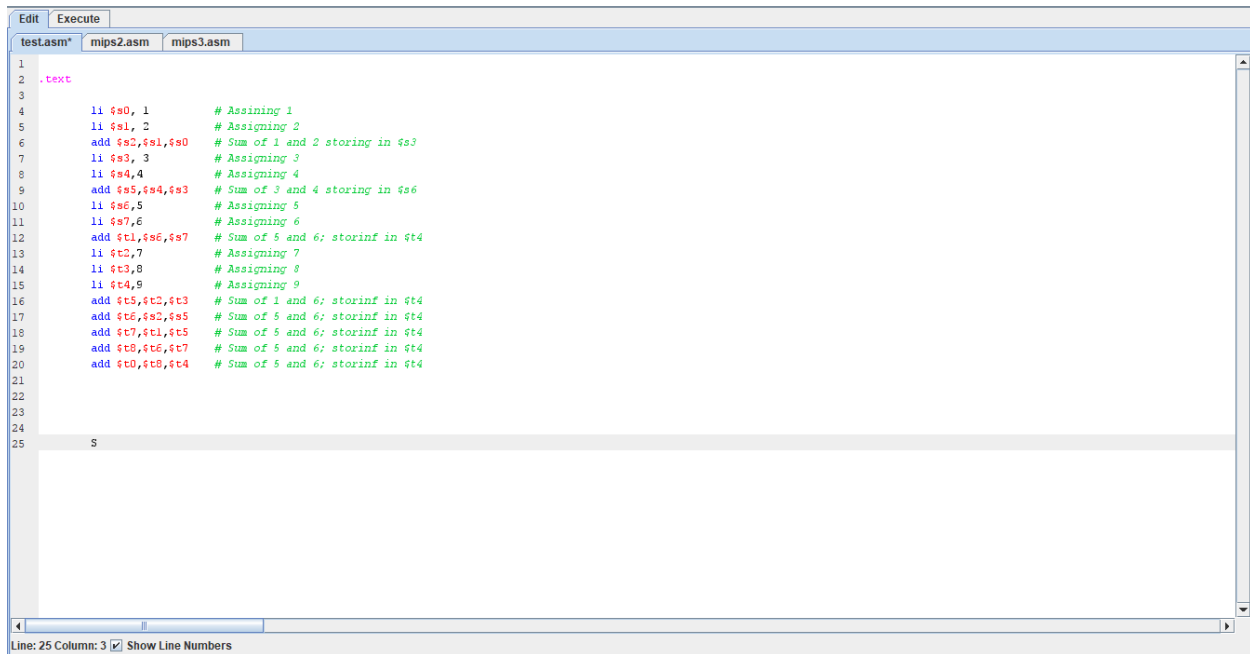
Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+C)	Value (+10)	Value (+14)	Value (+18)	Value (+1C)
0x10010000	0	0	0	0	0	0	0	0
0x10010004	0	0	0	0	0	0	0	0
0x10010008	0	0	0	0	0	0	0	0
0x1001000c	0	0	0	0	0	0	0	0
0x10010010	0	0	0	0	0	0	0	0
0x10010014	0	0	0	0	0	0	0	0
0x10010018	0	0	0	0	0	0	0	0
0x1001001c	0	0	0	0	0	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010024	0	0	0	0	0	0	0	0
0x10010028	0	0	0	0	0	0	0	0
0x1001002c	0	0	0	0	0	0	0	0
0x10010030	0	0	0	0	0	0	0	0
0x10010034	0	0	0	0	0	0	0	0
0x10010038	0	0	0	0	0	0	0	0
0x1001003c	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010044	0	0	0	0	0	0	0	0
0x10010048	0	0	0	0	0	0	0	0
0x1001004c	0	0	0	0	0	0	0	0
0x10010050	0	0	0	0	0	0	0	0
0x10010054	0	0	0	0	0	0	0	0
0x10010058	0	0	0	0	0	0	0	0
0x1001005c	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010064	0	0	0	0	0	0	0	0
0x10010068	0	0	0	0	0	0	0	0
0x1001006c	0	0	0	0	0	0	0	0
0x10010070	0	0	0	0	0	0	0	0
0x10010074	0	0	0	0	0	0	0	0
0x10010078	0	0	0	0	0	0	0	0
0x1001007c	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x10010084	0	0	0	0	0	0	0	0
0x10010088	0	0	0	0	0	0	0	0
0x1001008c	0	0	0	0	0	0	0	0
0x10010090	0	0	0	0	0	0	0	0
0x10010094	0	0	0	0	0	0	0	0
0x10010098	0	0	0	0	0	0	0	0
0x1001009c	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100a4	0	0	0	0	0	0	0	0
0x100100a8	0	0	0	0	0	0	0	0
0x100100ac	0	0	0	0	0	0	0	0
0x100100b0	0	0	0	0	0	0	0	0
0x100100b4	0	0	0	0	0	0	0	0
0x100100b8	0	0	0	0	0	0	0	0
0x100100bc	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100c4	0	0	0	0	0	0	0	0
0x100100c8	0	0	0	0	0	0	0	0
0x100100cc	0	0	0	0	0	0	0	0
0x100100d0	0	0	0	0	0	0	0	0
0x100100d4	0	0	0	0	0	0	0	0
0x100100d8	0	0	0	0	0	0	0	0
0x100100dc	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0
0x100100e4	0	0	0	0	0	0	0	0
0x100100e8	0	0	0	0	0	0	0	0
0x100100ec	0	0	0	0	0	0	0	0
0x100100f0	0	0	0	0	0	0	0	0
0x100100f4	0	0	0	0	0	0	0	0
0x100100f8	0	0	0	0	0	0	0	0
0x100100fc	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010104	0	0	0	0	0	0	0	0
0x10010108	0	0	0	0	0	0	0	0
0x1001010c	0	0	0	0	0	0	0	0
0x10010110	0	0	0	0	0	0	0	0
0x10010114	0	0	0	0	0	0	0	0
0x10010118	0	0	0	0	0	0	0	0
0x1001011c	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0
0x10010124	0	0	0	0	0	0	0	0
0x10010128	0	0	0	0	0	0	0	0
0x1001012c	0	0	0	0	0	0	0	0
0x10010130	0	0	0	0	0	0	0	0
0x10010134	0	0	0	0	0	0	0	0
0x10010138	0	0	0	0	0	0	0	0
0x1001013c	0	0	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0	0	0
0x10010144	0	0	0	0	0	0	0	0
0x10010148	0	0	0	0	0	0	0	0
0x1001014c	0	0	0	0	0	0	0	0
0x10010150	0	0	0	0	0	0	0	0
0x10010154	0	0	0	0	0	0	0	0
0x10010158	0	0	0	0	0	0	0	0
0x1001015c	0	0	0	0	0	0	0	0
0x10010160	0	0	0	0	0	0	0	0
0x10010164	0	0	0	0	0	0	0	0
0x10010168	0	0	0	0	0	0	0	0
0x1001016c	0	0	0	0	0	0	0	0
0x10010170	0	0	0	0	0	0	0	0
0x10010174	0	0	0	0	0	0	0	0
0x10010178	0	0	0	0	0	0	0	0
0x1001017c	0	0	0	0	0	0	0	0
0x10010180	0	0	0	0	0	0	0	0
0x10010184	0	0	0	0	0	0	0	0
0x10010188	0	0	0	0	0	0	0	0
0x1001018c	0	0	0	0	0	0	0	0
0x10010190	0	0	0	0	0	0	0	0
0x10010194	0	0	0	0	0	0	0	0
0x10010198	0	0	0	0	0	0	0	0
0x1001019c	0	0	0	0	0	0	0	0
0x100101a0	0	0	0	0	0	0	0	0
0x100101a4	0	0	0	0	0	0	0	0
0x100101a8	0	0	0	0	0	0	0	0
0x100101ac	0	0	0	0	0	0	0	0
0x100101b0	0	0	0	0	0	0	0	0
0x100101b4	0	0	0	0	0	0	0	0
0x100101b8	0	0	0	0	0	0	0	0
0x100101bc	0	0	0	0	0	0	0	0
0x100101c0	0	0	0	0	0	0	0	0
0x100101c4	0	0	0	0	0	0	0	0
0x100101c8	0	0	0	0	0	0	0	0
0x100101cc	0	0	0	0	0	0	0	0
0x100101d0	0	0	0	0	0	0	0	0
0x100101d4	0	0	0	0	0	0	0	0
0x100101d8	0	0	0	0	0	0	0	0
0x100101dc	0	0	0	0	0	0	0	0
0x100101e0	0	0	0	0	0	0	0	0
0x100101e4	0	0	0	0	0	0	0	0
0x100101e8	0	0	0	0	0	0	0	0
0x100101ec	0	0	0	0	0	0	0	0
0x100101f0	0	0	0	0	0	0	0	0
0x100101f4	0	0	0	0	0	0	0	0
0x100101f8	0	0	0	0	0	0	0	0
0x100101fc	0	0	0	0	0	0	0	0
0x10010200	0	0	0	0	0	0	0	0
0x10010204	0	0	0	0	0	0	0	0
0x10010208	0	0	0	0	0	0	0	0
0x1001020c	0	0	0	0	0	0	0	0
0x10010210	0	0	0	0	0	0	0	0
0x10010214	0	0	0	0	0	0	0	0
0x10010218	0	0	0	0	0	0	0	0
0x1001021c	0	0	0	0	0	0	0	0
0x10010220	0	0	0	0	0	0	0	0
0x10010224	0	0	0	0	0	0	0	0
0x10010228	0	0	0	0	0	0	0	0
0x1001022c	0	0	0	0	0	0	0	0
0x10010230	0	0	0	0	0	0	0	0
0x10010234	0	0	0	0	0	0	0	0
0x10010238	0	0	0	0	0	0	0	0
0x1001023c	0	0	0	0	0	0	0	0
0x10010240	0	0	0	0	0	0	0	0
0x10010244	0	0	0	0	0	0	0	0
0x10010248	0	0	0	0	0	0	0	0
0x1001024c	0	0	0	0	0	0	0	0
0x10010250	0	0	0	0	0	0	0	0
0x10010254	0	0	0	0	0	0	0	0
0x10010258	0	0	0	0	0	0	0	0
0x1001025c	0	0	0	0	0	0	0	0
0x10010260	0	0	0	0	0	0	0	0
0x10010264	0	0	0	0	0	0	0	0
0x10010268	0	0	0	0	0	0	0	0
0x1001026c	0	0	0	0	0	0	0	0
0x10010270	0	0	0	0	0	0	0	0
0x10010274	0	0	0	0	0	0	0	0
0x10010278	0	0	0	0	0	0	0	0
0x1001027c	0	0	0	0	0	0	0	0
0x10010280	0	0	0	0	0	0	0	0
0x10010284	0	0	0	0	0	0	0	0
0x10010288	0	0	0	0	0	0	0	0
0x1001028c	0	0	0	0	0	0	0	0
0x10010290	0	0	0	0	0	0	0	0
0x10010294	0	0	0	0	0	0	0	0
0x1001020								

## Problem 2:

Write a MIPS Assembly program to ADD the numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9.

- Use ONLY the instructions: add and li
- Use any \$t, \$s registers
- The result of the addition (in decimal) should be in register: \$t0



```
1
2 .text
3
4     li $s0, 1      # Assigning 1
5     li $s1, 2      # Assigning 2
6     add $s2,$s1,$s0 # Sum of 1 and 2 storing in $s3
7     li $s3, 3      # Assigning 3
8     li $s4, 4      # Assigning 4
9     add $s5,$s4,$s3 # Sum of 3 and 4 storing in $s6
10    li $s6, 5      # Assigning 5
11    li $s7, 6      # Assigning 6
12    add $t1,$s6,$s7 # Sum of 5 and 6; storing in $t4
13    li $t2, 7      # Assigning 7
14    li $t3, 8      # Assigning 8
15    li $t4, 9      # Assigning 9
16    add $t5,$t2,$t3 # Sum of 1 and 6; storing in $t4
17    add $t6,$s2,$s5 # Sum of 5 and 6; storing in $t4
18    add $t7,$t1,$t5 # Sum of 5 and 6; storing in $t4
19    add $t8,$t6,$t7 # Sum of 5 and 6; storing in $t4
20    add $t0,$t8,$t4 # Sum of 5 and 6; storing in $t4
21
22
23
24
25    S
```

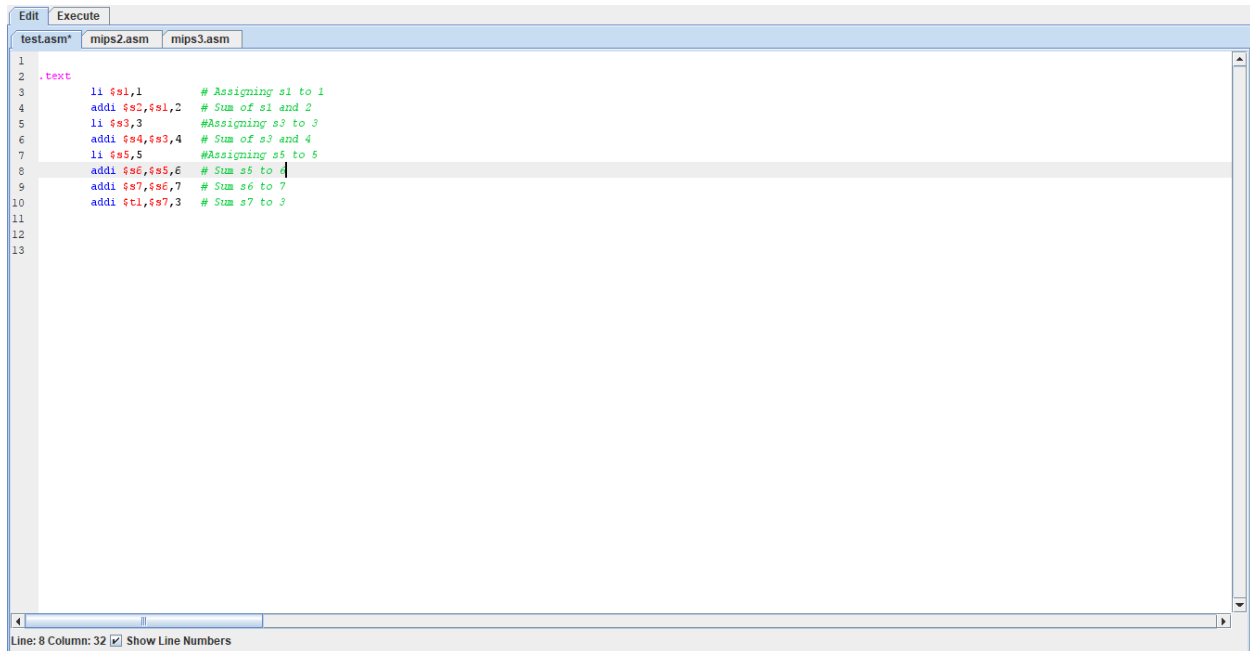
Line: 25 Column: 3 ☒ Show Line Numbers

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	45
\$t1	9	11
\$t2	10	7
\$t3	11	8
\$t4	12	9
\$t5	13	15
\$t6	14	10
\$t7	15	26
\$s0	16	1
\$s1	17	2
\$s2	18	3
\$s3	19	3
\$s4	20	4
\$s5	21	7
\$s6	22	5
\$s7	23	6
\$t8	24	36
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194372
hi		0
lo		0

### Problem 3:

Write a MIPS Assembly program to ADD the numbers: 1, 2, 3, 4, 5, 6 (as in ex-2)

- Use ONLY the instructions: addi and li
- The result of the addition (in decimal) should be in register: \$t1



The screenshot shows a MIPS assembly editor window with the following code:

```
1  
2 .text  
3 li $s1,1      # Assigning s1 to 1  
4 addi $s2,$s1,2 # Sum of s1 and 2  
5 li $s3,3      # Assigning s3 to 3  
6 addi $s4,$s3,4 # Sum of s3 and 4  
7 li $s5,5      # Assigning s5 to 5  
8 addi $s6,$s5,6 # Sum s5 to 6  
9 addi $s7,$s6,7 # Sum s6 to 7  
10 addi $t1,$s7,3 # Sum s7 to 3  
11  
12  
13
```

The status bar at the bottom indicates "Line: 8 Column: 32" and "Show Line Numbers" is checked.

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	21
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	1
\$s2	18	3
\$s3	19	3
\$s4	20	7
\$s5	21	5
\$s6	22	11
\$s7	23	18
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194336
hi		0
lo		0

## Problem 4:

Write a MIPS Assembly program to ADD the numbers: 1, 2, 3, 4, 5, 6 (as in ex-3)

- Use ONLY the instructions: addi and li ... and ONLY the register: \$t0
- The result of the addition (in decimal) should be in register: \$t0

EditExecute

testasm\*

mips2.asm

mips3.asm

```

1
2 .text
3 li $t0, 1          #Assigned value to $t0
4 addi $t0, $t0, 2   # Adding 2 to $t0
5 addi $t0, $t0, 3   # Adding 3 to $t0
6 addi $t0, $t0, 4   # Adding 4 to $t0
7 addi $t0, $t0, 5   # Adding 5 to $t0
8 addi $t0, $t0, 6   # Adding 6 to $t0
9
10
11
12
13

```

Line: 8 Column: 29 ☒ Show Line Numbers

EditExecute

Registers

Coproc 1

Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
<b>\$t0</b>	<b>8</b>	<b>21</b>
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$f0	26	0
\$f1	27	0
\$fp	28	269469224
\$gp	29	2147479548
\$fp	30	0
\$ra	31	0
\$pc		4194360
\$j0		0