

Portfolio 2 Report

EEL3801C - C011

Yousef Awad

June 2025

Contents

1a	2
code	2
output	3
1b	4
code	4
output	6
2a	7
code	7
output	8
2b	9
code	9
output	10

1a

code

add.asm	counter.asm	counter-hexa.asm	add-showinput.asm
<pre>1 # Yousef Awad -- 06/02/2025 2 # 3 # add.asm 4 # 5 # A program that computes 6 # and prints the 7 # summation of three 8 # numbers specficed at 9 # runtime by the user. 10 # 11 # Registers Used: 12 # \$t0 - used to hold the first number 13 # \$t1 - used to hold the second number 14 # \$t2 - used to hold the third number 15 # \$t3 - used to hold the sum 16 # \$v0 - syscall parameter and return value 17 # \$a0 - syscall parameter (print value) 18 19 main: 20 21 # Getting first number (\$t0) from user 22 li \$v0, 5 # 5 signifies read_int 23 syscall 24 move \$t0, \$v0 25 26 # Getting second number (\$t1) from user. 27 li \$v0, 5 28 syscall 29 move \$t1, \$v0 30 31 # Getting third number (\$t2) from user. 32 li \$v0, 5 33 syscall 34 move \$t2, \$v0 35 36 # Computing triple sum 37 add \$t3, \$t0, \$t1 38 add \$t3, \$t2, \$t3 39 40 # Printing out \$t3 (sum) 41 li \$v0, 1 42 la \$a0, (\$t3) 43 syscall 44 45 # Ending program 46 li \$v0, 10 47 syscall</pre>			

output

Bkpt	Address	Code	Basic
<input type="checkbox"/>	0x00400000	0x24020005	addiu \$2,\$0,0x00000005
<input type="checkbox"/>	0x00400004	0x0000000c	syscall
<input type="checkbox"/>	0x00400008	0x00024021	addu \$8,\$0,\$2
<input type="checkbox"/>	0x0040000c	0x24020005	addiu \$2,\$0,0x00000005
<input type="checkbox"/>	0x00400010	0x0000000c	syscall
<input type="checkbox"/>	0x00400014	0x00024821	addu \$0,\$0,\$2
<input type="checkbox"/>	0x00400018	0x24020005	addiu \$2,\$0,0x00000005
<input type="checkbox"/>	0x0040001c	0x0000000c	syscall
<input type="checkbox"/>	0x00400020	0x00025021	addu \$10,\$0,\$2
<input type="checkbox"/>	0x00400024	0x01095820	add \$11,\$8,\$9
<input type="checkbox"/>	0x00400028	0x014b5820	add \$11,\$10,\$11
<input type="checkbox"/>	0x0040002c	0x24020001	addiu \$2,\$0,0x00000001
<input type="checkbox"/>	0x00400030	0x21640000	addi \$4,\$11,0x00000000
<input type="checkbox"/>	0x00400034	0x0000000c	syscall
<input type="checkbox"/>	0x00400038	0x2402000a	addiu \$2,\$0,0x0000000a
<input type="checkbox"/>	0x0040003c	0x0000000c	syscall

Address	Value (+0)	Value (+4)
0x10010000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000

Mars Messages	Run I/O
1	
2	
3	
6	
-- program is finished running --	

1b

code

add.asm	counter.asm	counter-hexa.asm	add-showinput.asm
---------	-------------	------------------	-------------------

```
1  # Yousef Awad -- 06/02/2025
2  #
3  # add-showinput.asm
4  #
5  # A program that computes
6  # and prints the
7  # summation of three
8  # numbers specified at
9  # runtime by the user.
10 #
11 # Program also shows
12 # the user's inputted
13 # three numbers unlike
14 # add.asm
15 #
16 # Registers Used:
17 # $t0 - used to hold the first number
18 # $t1 - used to hold the second number
19 # $t2 - used to hold the third number
20 # $t3 - used to hold the sum
21 # $v0 - syscall parameter and return value
22 # $a0 - syscall parameter (print value)
23
24 .data
25 # holds user inputted string
26 number1: .space 1
27 number2: .space 1
28 number3: .space 1
29 plus: .asciiz "+"
30 equal: .asciiz "="
31 .text
32 |
33 main:
34
35 # Getting first number ($t0) from user
36 li $v0, 5 # 5 signifies read_int
37 syscall
38 move $t0, $v0
39
40 # Getting second number ($t1) from user.
41 li $v0, 5
42 syscall
43 move $t1, $v0
44
45 # Getting third number ($t2) from user.
46 li $v0, 5
47 syscall
48 move $t2, $v0
49
50 # Printing out user input
51 #
52 # Printing first number
53 li $v0, 1
54 la $a0, ($t0)
55 syscall
56 # Printing '+'
57 li $v0, 4
58 la $a0, plus
59 syscall
60 # Printing second number
61 li $v0, 1
62 la $a0, ($t1)
63 syscall
64 # Printing '+'
65 li $v0, 4
66 la $a0, plus
67 syscall
```

add.asm	counter.asm	counter-hexa.asm	add-showinput.asm
---------	-------------	------------------	-------------------

```

24 .data
25 # holds user inputted string
26 number1: .space 1
27 number2: .space 1
28 number3: .space 1
29 plus: .asciiz "+"
30 equal: .asciiz "="
31 .text
32
33 main:
34
35 # Getting first number ($t0) from user
36 li $v0, 5 # 5 signifies read_int
37 syscall
38 move $t0, $v0
39
40 # Getting second number ($t1) from user.
41 li $v0, 5
42 syscall
43 move $t1, $v0
44
45 # Getting third number ($t2) from user.
46 li $v0, 5
47 syscall
48 move $t2, $v0
49
50 # Printing out user input
51 #
52 # Printing first number
53 li $v0, 1
54 la $a0, ($t0)
55 syscall
56 # Printing '+'
57 li $v0, 4
58 la $a0, plus
59 syscall
60 # Printing second number
61 li $v0, 1
62 la $a0, ($t1)
63 syscall
64 # Printing '+'
65 li $v0, 4
66 la $a0, plus
67 syscall
68 # Printing third number
69 li $v0, 1
70 la $a0, ($t2)
71 syscall
72 # Printing '='
73 li $v0, 4
74 la $a0, equal
75 syscall
76 # Printing newline
77 li $v0, 11
78 li $a0, 10
79 syscall
80
81 # Computing triple sum
82 add $t3, $t0, $t1
83 add $t3, $t2, $t3
84
85 # Printing out $t3 (sum)
86 li $v0, 1
87 la $a0, ($t3)
88 syscall
89
90 # Ending program
91 li $v0, 10
92 syscall

```

output

Text Segment

Bkpt	Address	Code	Basic	
<input type="checkbox"/>	0x00400000	0x24020005	addiu \$2,\$0,0x00000005	36: li \$v0, 5 # 5 signifies read_int
<input type="checkbox"/>	0x00400004	0x0000000c	syscall	37: syscall
<input type="checkbox"/>	0x00400008	0x00024021	addu \$8,\$0,\$2	38: move \$t0, \$v0
<input type="checkbox"/>	0x0040000c	0x24020005	addiu \$2,\$0,0x00000005	41: li \$v0, 5
<input type="checkbox"/>	0x00400010	0x0000000c	syscall	42: syscall
<input type="checkbox"/>	0x00400014	0x00024021	addu \$9,\$0,\$2	43: move \$t1, \$v0
<input type="checkbox"/>	0x00400018	0x24020005	addiu \$2,\$0,0x00000005	46: li \$v0, 5
<input type="checkbox"/>	0x0040001c	0x0000000c	syscall	47: syscall
<input type="checkbox"/>	0x00400020	0x00025021	addu \$10,\$0,\$2	48: move \$t2, \$v0
<input type="checkbox"/>	0x00400024	0x24020001	addiu \$2,\$0,0x00000001	53: li \$v0, 1
<input type="checkbox"/>	0x00400028	0x21040000	addi \$4,\$8,0x00000000	54: la \$a0, (\$t0)
<input type="checkbox"/>	0x0040002c	0x0000000c	syscall	55: syscall
<input type="checkbox"/>	0x00400030	0x24020004	addiu \$2,\$0,0x00000004	57: li \$v0, 4
<input type="checkbox"/>	0x00400034	0x3c011001	lui \$1,0x00001001	58: la \$a0, plus
<input type="checkbox"/>	0x00400038	0x34240003	ori \$4,\$1,0x00000003	
<input type="checkbox"/>	0x0040003c	0x0000000c	syscall	59: syscall
<input type="checkbox"/>	0x00400040	0x24020001	addiu \$2,\$0,0x00000001	61: li \$v0, 1
<input type="checkbox"/>	0x00400044	0x21240000	addi \$4,\$9,0x00000000	62: la \$a0, (\$t1)
<input type="checkbox"/>	0x00400048	0x0000000c	syscall	63: syscall
<input type="checkbox"/>	0x0040004c	0x24020004	addiu \$2,\$0,0x00000004	65: li \$v0, 4
<input type="checkbox"/>	0x00400050	0x3c011001	lui \$1,0x00001001	66: la \$a0, plus
<input type="checkbox"/>	0x00400054	0x34240003	ori \$4,\$1,0x00000003	
<input type="checkbox"/>	0x00400058	0x0000000c	syscall	67: syscall
<input type="checkbox"/>	0x0040005c	0x24020001	addiu \$2,\$0,0x00000001	69: li \$v0, 1
<input type="checkbox"/>	0x00400060	0x21440000	addi \$4,\$10,0x00000000	70: la \$a0, (\$t2)
<input type="checkbox"/>	0x00400064	0x0000000c	syscall	71: syscall
<input type="checkbox"/>	0x00400068	0x24020004	addiu \$2,\$0,0x00000004	73: li \$v0, 4

Data Segment

Address	Value (+0)	Value (+4)	Value
0x10010000	0x2b000000	0x00003d00	
0x10010020	0x00000000	0x00000000	
0x10010040	0x00000000	0x00000000	
0x10010060	0x00000000	0x00000000	
0x10010080	0x00000000	0x00000000	
0x100100a0	0x00000000	0x00000000	
0x100100c0	0x00000000	0x00000000	
0x100100e0	0x00000000	0x00000000	
0x10010100	0x00000000	0x00000000	
0x10010120	0x00000000	0x00000000	
0x10010140	0x00000000	0x00000000	
0x10010160	0x00000000	0x00000000	
0x10010180	0x00000000	0x00000000	
0x100101a0	0x00000000	0x00000000	
0x100101c0	0x00000000	0x00000000	
0x100101e0	0x00000000	0x00000000	

Mars Messages

Run I/O

```

1
2
3
1+2+3=
6
-- program is finished running --

```

2a

code

add.asm	counter.asm	counter-hexa.asm	add-showinput.asm
---------	-------------	------------------	-------------------

```
1  # Yousef Awad -- 06/02/2025
2  #
3  # counter.asm
4  #
5  # A program that counts
6  # from 5 to 20 while
7  # displaying such to
8  # the user in binary.
9  #
10 # Registers Used:
11 # $t0 - Current Number
12 # $t1 - Max Number
13 # $v0 - Syscall/Return Parameter
14 # $a0 - Syscall Parameter (print)
15
16 main:
17
18 # initializing current and max
19 li $t0, 5
20 li $t1, 21 # must be one greater than wanted value
21
22 # loop proper
23 loop: bge $t0, $t1, exit
24
25     # printing current number ($t0)
26     li $v0, 1
27     la $a0, ($t0)
28     syscall
29
30     # printing newline
31     li $v0, 11
32     li $a0, 10
33     syscall
34
35     # incrementing current by 1
36     addi $t0, $t0, 1
37
38     # jump to check if $t0 > $t1 - 1
39     j loop
40
41 # exit branch (if $t0 > $t1 - 1)
42 exit:
43 li $v0, 10
44 syscall
```

output

Text Segment			
Bkpt	Address	Code	Basic
	0x00400000	0x24080005 addiu \$8,\$0,0x00000005	19: li \$t0, 5
	0x00400004	0x24090015 addiu \$9,\$0,0x00000015	20: li \$t1, 21 # must be one greater than wanted value
	0x00400008	0x0109082a slt \$l,\$8,\$9	23: loop: bge \$t0, \$t1, exit
	0x0040000c	0x10200008 beq \$l,\$0,0x00000008	
	0x00400010	0x24020001 addiu \$2,\$0,0x00000001	26: li \$v0, 1
	0x00400014	0x21040000 addi \$4,\$8,0x00000000	27: la \$a0, (\$t0)
	0x00400018	0x0000000c syscall	28: syscall
	0x0040001c	0x2402000b addiu \$2,\$0,0x0000000b	31: li \$v0, 11
	0x00400020	0x2404000a addiu \$4,\$0,0x0000000a	32: li \$a0, 10
	0x00400024	0x0000000c syscall	33: syscall
	0x00400028	0x21080001 addi \$8,\$8,0x00000001	36: addi \$t0, \$t0, 1
	0x0040002c	0x08100002 j 0x00400008	39: j loop
	0x00400030	0x2402000a addiu \$2,\$0,0x0000000a	43: li \$v0, 10
	0x00400034	0x0000000c syscall	44: syscall

Data Segment			
Address	Value (+0)	Value (+4)	Value (+8)
0x10010000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000	0x00000000

◀ ▶ 0x10010000 (da

Mars Messages Run I/O

```

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

-- program is finished running --

```

Clear

2b

code

add.asm	counter.asm	counter-hexa.asm	add-showinput.asm
<pre>1 # Yousef Awad -- 06/02/2025 2 # 3 # counter-hexa.asm 4 # 5 # A program that counts 6 # from 5 to 20 while 7 # displaying such to 8 # the user in hexadecimal. 9 # 10 # Registers Used: 11 # \$t0 - Current Number 12 # \$t1 - Max Number 13 # \$v0 - Syscall/Return Parameter 14 # \$a0 - Syscall Parameter (print) 15 16 main: 17 18 # initializing current and max 19 li \$t0, 5 20 li \$t1, 21 # must be one greater than wanted value 21 22 # loop proper 23 loop: bge \$t0, \$t1, exit 24 25 # printing current number (\$t0) 26 li \$v0, 34 # code for printing in hexa 27 la \$a0, (\$t0) 28 syscall 29 30 # printing newline 31 li \$v0, 11 32 li \$a0, 10 33 syscall 34 35 # incrementing current by 1 36 addi \$t0, \$t0, 1 37 38 # jump to check if \$t0 > \$t1 - 1 39 j loop 40 41 # exit branch (if \$t0 > \$t1 - 1) 42 exit: 43 li \$v0, 10 44 syscall</pre>			

output

Text Segment

Bkpt	Address	Code	Basic	
	0x00400000	0x24080005	addiu \$8,\$0,0x00000005	19: li \$t0, 5
	0x00400004	0x24090015	addiu \$9,\$0,0x00000015	20: li \$t1, 21 # must be one greater than wanted value
	0x00400008	0x0109082a	slt \$1,\$8,\$9	23: loop: bge \$t0, \$t1, exit
	0x0040000c	0x10200008	beq \$1,\$0,0x00000008	
	0x00400010	0x24020022	addiu \$2,\$0,0x00000022	26: li \$v0, 34 # code for printing in hexa
	0x00400014	0x21040000	addi \$4,\$8,0x00000000	27: la \$a0, (\$t0)
	0x00400018	0x0000000c	syscall	28: syscall
	0x0040001c	0x2402000b	addiu \$2,\$0,0x0000000b	31: li \$v0, 11
	0x00400020	0x2404000a	addiu \$4,\$0,0x0000000a	32: li \$a0, 10
	0x00400024	0x0000000c	syscall	33: syscall
	0x00400028	0x21080001	addi \$8,\$8,0x00000001	36: addi \$t0, \$t0, 1
	0x0040002c	0x08100002	j 0x00400008	39: j loop
	0x00400030	0x2402000a	addiu \$2,\$0,0x0000000a	43: li \$v0, 10
	0x00400034	0x0000000c	syscall	44: syscall

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)
0x10010000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000
0x100101e0	0x00000000	0x00000000	0x00000000

0x10010000 {, da

Mars Messages

Run I/O

Clear

0x00000005
0x00000006
0x00000007
0x00000008
0x00000009
0x0000000a
0x0000000b
0x0000000c
0x0000000d
0x0000000e
0x0000000f
0x00000010
0x00000011
0x00000012
0x00000013
0x00000014

-- program is finished running --