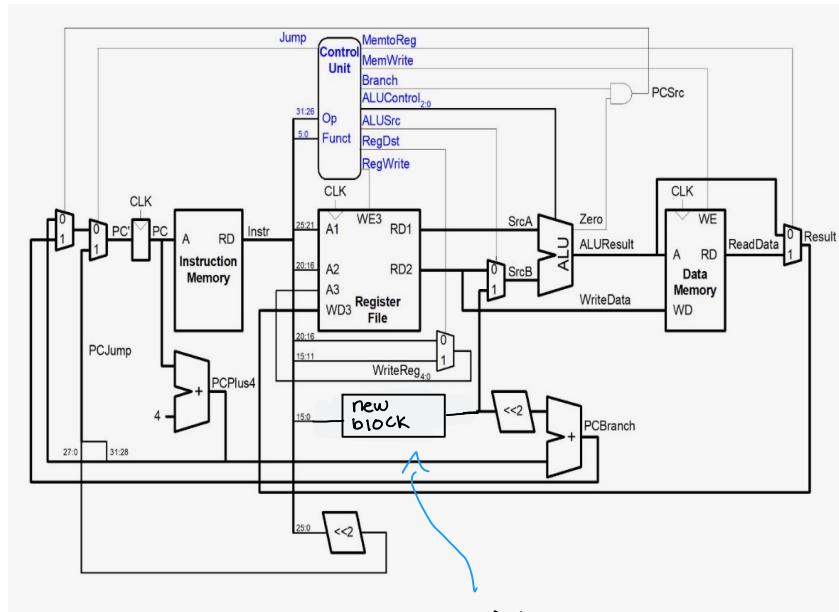


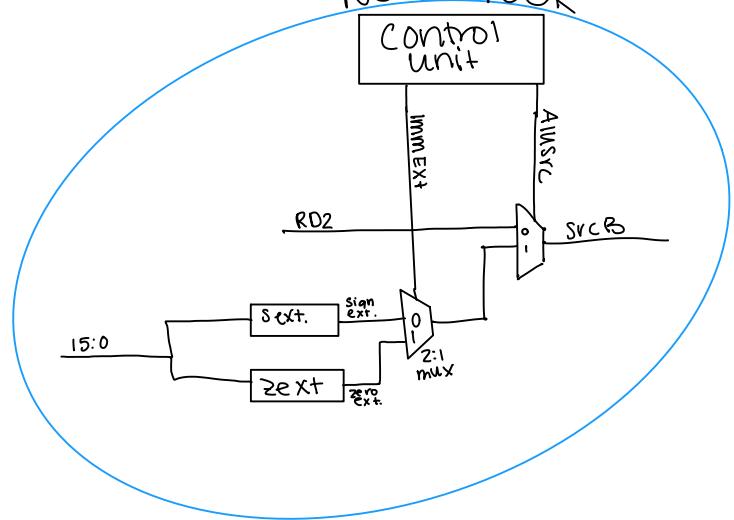
Homework 4

Friday, July 29, 2022 4:01 PM

Q1



NEW block

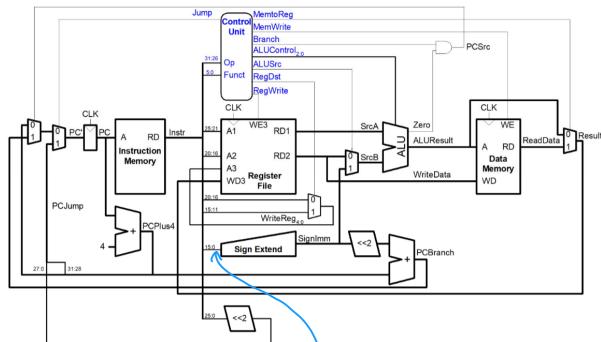


Control Unit- Instruction Decoder 2

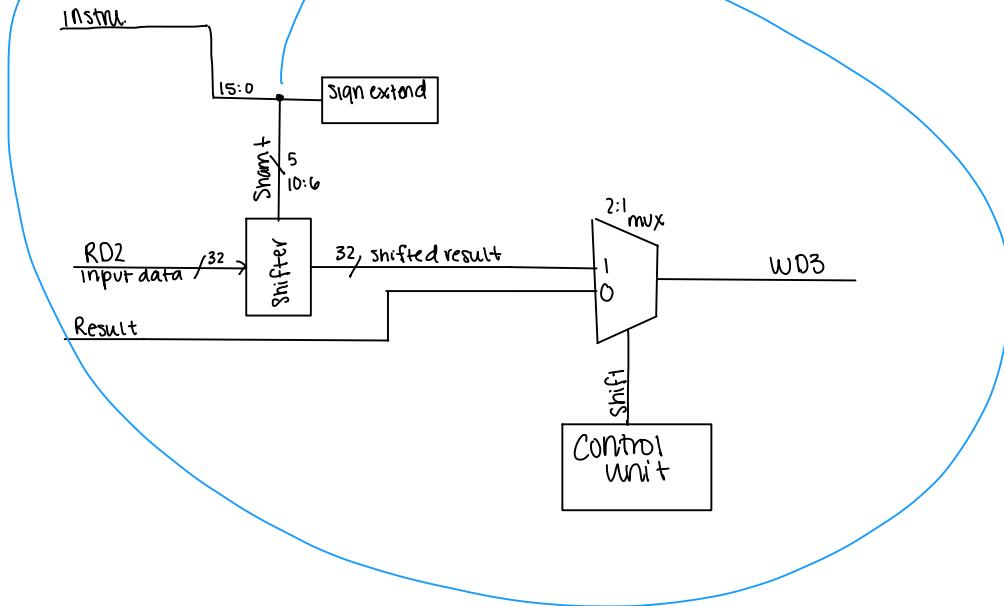
Instruction	Op[5:0]	RegWrite	RegDst	ALUSrc	Branch	MemWrite	MemtoReg	ALUControl[2:0]	Jump	ImmExtend
R-type	000000	1	1	0	0	0	0	Depends func	0	0
lw	100011	1	0	1	0	0	1	010	0	0
sw	101011	0	x	1	0	1	x	010	0	0
beq	000100	0	x	0	1	0	x	110	0	0
addi	001000	1	0	1	0	0	0	010	0	0
j	000010	0	x	x	x	0	x	xxx	1	x
ori	001101	1	0	1	0	0	0	001	0	1

Q2

S11 performs a mult operation on both Signed & unsigned
 R type \Rightarrow shamt [10:6]



new block:



Q3 Find $T_c \rightarrow$ amt time for 1 cycle
 * no need to consider setup time for regfile/datamem
 * determined by critical path.

$$(i) \text{ add: } T_{pq} + t_{mem} + \max(t_{RF}, t_{mux}, t_{control}, t_{mux}) + t_{ALU} + t_{mux}$$

$$30 + 250 + 150 + 200 + 25 = \textcircled{655}$$

$$(ii) \text{ SW: } T_{pq} + t_{mem} + \max(t_{RF}, t_{mux}, t_{control}, t_{mux}) + t_{ALU}$$

$$30 + 250 + 150 + 200 + 250 = \textcircled{880}$$

Q4: Screenshot of Output with input string: "Hello my name is Kaite"
 Output: hellomynameiskaite



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20020004	addi \$2,\$0,0x00000004	11: addi \$v0,\$0,4
	0x00400004	0x3c011001	lui \$1,0x00001001	12: la \$a0,ask
	0x00400008	0x34240000	ori \$4,\$1,0x00000000	
	0x0040000c	0x0000000c	syscall	13: syscall
	0x00400010	0x20020008	addi \$2,\$0,0x00000008	15: addi \$v0,\$zero,8
	0x00400014	0x3c011001	lui \$1,0x00001001	16: la \$a0,length
	0x00400018	0x34240016	ori \$4,\$1,0x00000016	
	0x0040001c	0x24050064	addiu \$5,\$0,0x00000064	17: li \$a1,100
	0x00400020	0x0000000c	syscall	18: syscall
	0x00400024	0x20080000	addi \$8,\$0,0x00000000	20: addi \$t0,\$0,0
	0x00400028	0x20090000	addi \$9,\$0,0x00000000	21: addi \$t1,\$0,0
	0x0040002c	0x3c011001	lui \$1,0x00001001	23: loop: lb \$t2,length(\$t0)
	0x00400030	0x00280821	addu \$1,\$1,\$8	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	a e l P	e e s	r e t n	r t s	: g n i	e h \0	m o l l	m a n y
0x10010020	k s i e	e t i a	e t \0 \n	\0 \0 \0 \n	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	g n i r	\0 :	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010120	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010140	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

Mars Messages Run I/O

```
-- program is finished running --

Please enter string: hello my name is kaite
New String: hellomynameiskaite
Clear
-- program is finished running --
```

Q5:Screenshots of outputs with different arrays and different X values

i) Test1: A: 1 2 3 4 4 4 5 5 6 1

X=4, F=3

X=5, F=2

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1,0x00001001	13: la \$a1,A
	0x00400004	0x34250032	ori \$5,\$1,0x00000032	
	0x00400008	0x3c011001	lui \$1,0x00001001	14: lbu \$a2,N
	0x0040000c	0x90260046	lbu \$6,0x00000046(\$1)	#load the value of N (size) to \$a2
	0x00400010	0x3c011001	lui \$1,0x00001001	16: la \$a0,ask
	0x00400014	0x34240000	ori \$4,\$1,0x00000000	#ask user to input N
	0x00400018	0x20020004	addi \$2,\$0,0x00000004	17: addi \$v0,\$0,4
	0x0040001c	0x0000000c	syscall	18: syscall
	0x00400020	0x20020005	addi \$2,\$0,0x00000005	20: addi \$v0,\$0,5
	0x00400024	0x0000000c	syscall	21: syscall
	0x00400028	0x20470000	addi \$7,\$2,0x00000000	23: addi \$a3,\$v0,0
	0x0040002c	0xc100013	jal 0x0040004c	#loading user input stored in \$v0 to \$a3 as pa...
	0x00400030	0x3c011001	lui \$1,0x00001001	25: jal numoccur
				#call occurrences func
				27: sb \$v0,F

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	e t n E	a r	u l a v	o t e	n i f	h t d	u n e	r e b m
0x10010020	f o	u c c o	n a r r	: s e c	\0 . \0	\0 . \0 .	\0 . \0 .	\0 . \0 .
0x10010040	\0 . \0 .	. \n \0 .	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

Mars Messages Run I/O

```
Enter a value to find the number of occurrences: 4
3
-- program is finished running --

Clear
Reset: reset completed.

Enter a value to find the number of occurrences: 5
2
-- program is finished running --
```

ii) Test 2: A: 1 2 5 6 7 2 5 8 9 3 4

X=2, F=2
X=67, F=1

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1,0x00001001	13: la \$a1,A
	0x00400004	0x34250032	ori \$5,\$1,0x00000032	#load address of the array to \$a1
	0x00400008	0x3c011001	lui \$1,0x00001001	14: lbu \$a2,N
	0x0040000c	0x90260046	lbu \$6,0x00000046(\$1)	#load the value of N (size) to \$a2
	0x00400010	0x3c011001	lui \$1,0x00001001	16: la \$a0,ask
	0x00400014	0x34240004	ori \$4,\$1,0x00000004	#ask user to input N
	0x00400018	0x20020004	addi \$2,\$0,0x00000004	17: addi \$v0,\$0,4
	0x0040001c	0x00000000c	syscall	18: syscall
	0x00400020	0x20020005	addi \$2,\$0,0x00000005	20: addi \$v0,\$0,5
	0x00400024	0x00000000c	syscall	21: syscall
	0x00400028	0x20470000	addi \$7,\$2,0x00000000	23: addi \$a3,\$v0,0
	0x0040002c	0xc100013	jal 0x040004c	#loading user input stored in \$v0 to \$a3 as pa...
	0x00400030	0x3c011001	lui \$1,0x00001001	25: jal numoccur
				#call occurrences func
				27: sb \$v0,F
				#store F in \$v0

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	e t n E	a r	u l a v	o t e	n i f	h t d	u n e	r e b m
0x10010020	f o	u c c o	n a r r	: s e c	\0 . \0	\0 . \0 -	\0 . \0 .	\0 . \0 .
0x10010040	\0 . \0 .	. \n \0 \t	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages

```

Reset: reset completed.

Enter a value to find the number of occurrences: 2
2
-- program is finished running --

Reset: reset completed.

Enter a value to find the number of occurrences: 67
1

```

Test 3: Includes no occurrence

A=2 3 5 6 3 4 8 9 9 0
X = 1, F=0
X=3, F=2

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1,0x00001001	13: la \$a1,A
	0x00400004	0x34250032	ori \$5,\$1,0x00000032	#load address of the array to \$a1
	0x00400008	0x3c011001	lui \$1,0x00001001	14: lbu \$a2,N
	0x0040000c	0x90260046	lbu \$6,0x00000046(\$1)	#load the value of N (size) to \$a2
	0x00400010	0x3c011001	lui \$1,0x00001001	16: la \$a0,ask
	0x00400014	0x34240004	ori \$4,\$1,0x00000004	#ask user to input N
	0x00400018	0x20020004	addi \$2,\$0,0x00000004	17: addi \$v0,\$0,4
	0x0040001c	0x00000000c	syscall	18: syscall
	0x00400020	0x20020005	addi \$2,\$0,0x00000005	20: addi \$v0,\$0,5
	0x00400024	0x00000000c	syscall	21: syscall
	0x00400028	0x20470000	addi \$7,\$2,0x00000000	23: addi \$a3,\$v0,0
	0x0040002c	0xc100013	jal 0x040004c	#loading user input stored in \$v0 to \$a3 as pa...
	0x00400030	0x3c011001	lui \$1,0x00001001	25: jal numoccur
				#call occurrences func
				27: sb \$v0,F

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	e t n E	a r	u l a v	o t e	\0 . \0	\0 . \0 .	\0 . \0 .	\0 . \0 .
0x10010020	f o	u c c o	n a r r	: s e c	\0 . \0	\0 . \0 \0	\0 . \0 \0	\0 . \0 \0
0x10010040	\0 . \0 .	. \n \0 \t	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages

```

1
-- program is finished running --

Enter a value to find the number of occurrences: 0
0
-- program is finished running --

Reset: reset completed.

Enter a value to find the number of occurrences: 3

```

Q6: Screenshots of outputs with different arrays and array sizes.

i) Array size = 5

A: 1, 3, 5, 7, 9

The screenshot shows the Mars Simulation Environment interface. The assembly code window displays the following instructions:

```

    .Text
    .Basic
    9: la $s0, A           #load address of array store in $s0
    10: lw $s1, N($s0)     #load the size of array in $s1
    12: addi $s1, $s1, -1  #incrementing size of array backwards
    13: addi $sp, $sp, -12 #reserving space in the stack
    14: sw $s0, 0($sp)     #storing array in stack so its not altered by ...
    15: sw $s1, 4($sp)     #storing the size of array in stack so its not...
    16: sw $ra, 8($sp)     #storing the return address in stack so its no...
    18: jal reverse        #call reverse func
    20: lw $s0, 0($sp)     #retrieving values from the stack
    21: lw $s1, 4($sp)
  
```

The memory dump window shows the data segment with the following values:

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 \0 .	\0 \0 \0 .	\0 \0 \0 \t	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\0 \0 \0 \0 \b	\0 \0 \0 \0 \t	\0 \0 \0 \0 \n	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0
0x10010040	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0
0x10010060	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0
0x10010080	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0

The Mars Messages window shows the output of the program:

```

Enter a value to find the number of occurrences: 5
1
-- program is finished running --

9 7 5 3 1
-- program is finished running --
  
```

ii) Array size = 10

A: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

The screenshot shows the Mars Simulation Environment interface. The assembly code window displays the same set of instructions as the first screenshot, but with a larger memory dump. The memory dump window shows the data segment with the following values:

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 \0 .	\0 \0 \0 \0 .	\0 \0 \0 \0 .	\0 \0 \0 \0 .	\0 \0 \0 \0 .	\0 \0 \0 \0 .	\0 \0 \0 \0 .
0x10010020	\0 \0 \0 \0 \0 \b	\0 \0 \0 \0 \0 \t	\0 \0 \0 \0 \0 \n	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0
0x10010040	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0
0x10010060	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0
0x10010080	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0	\0 \0 \0 \0 \0 \0

The Mars Messages window shows the output of the program:

```

Enter a value to find the number of occurrences: 5
1
-- program is finished running --

9 7 5 3 1
-- program is finished running --

9 8 7 6 5 4 3 2 1 0
-- program is finished running --
  
```

iii) Array size = 15

A: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1,0x00001001	9: la \$s0, A #load address of array store in \$s0
	0x00400004	0x34300000	ori \$16,\$1,0x00000000	
	0x00400008	0x3c011001	lui \$1,0x00001001	10: lw \$s1, N(\$0) #load the size of array in \$s1
	0x0040000c	0x00200821	addu \$1,\$1,\$0	
	0x00400010	0x8c31003c	lw \$17,0x0000003c(\$1)	
	0x00400014	0x2231ffff	addi \$17,\$17,0xffff...	12: addi \$s1, \$s1, -1 #incrementing size of array backwards
	0x00400018	0x23bdfff4	addi \$29,\$29,0xffff...	13: addi \$sp, \$sp, -12 #reserving space in the stack
	0x0040001c	0xafb00000	sw \$16,0x00000000(\$29)	14: sw \$s0, 0(\$sp) #storing array in stack so its not altered by ...
	0x00400020	0xafb10004	sw \$17,0x00000004(\$29)	15: sw \$s1, 4(\$sp) #storing the size of array in stack so its not...
	0x00400024	0xafbf0008	sw \$31,0x00000008(\$29)	16: sw \$ra, 8(\$sp) #storing the return address in stack so its no...
	0x00400028	0x0c100012	jal 0x00400048	18: jal reverse #call reverse func
	0x0040002c	0x8fb00000	lw \$16,0x00000000(\$29)	20: lw \$s0, 0(\$sp) #retrieving values from the stack
	0x00400030	0x8fb10004	lw \$17,0x00000004(\$29)	21: lw \$s1, 4(\$sp)

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 \0 \0 .	\0 \0 \0 .	\0 \0 \0 .	\0 \0 \0 .	\0 \0 \0 .	\0 \0 \0 .
0x10010020	\0 \0 \0 \0 \b	\0 \0 \0 \0 \t	\0 \0 \0 \0 \n	\0 \0 \0 \0 \v	\0 \0 \0 \0 \f	\0 \0 \0 \0 \r	\0 \0 \0 \0 .	\0 \0 \0 \0 .
0x10010040	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0
0x10010060	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0
0x10010080	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0	\0 \0 \0 \0 \0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages Run I/O

```
-- program is finished running --
9 7 5 3 1
-- program is finished running --

Clear
9 8 7 6 5 4 3 2 1 0
-- program is finished running --

14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
-- program is finished running --
```

Name	Value
\$zero	
\$at	
\$v0	
\$v1	
\$a0	
\$a1	
\$a2	
\$a3	
\$t0	
\$t1	
\$t2	
\$t3	
\$t4	
\$t5	
\$t6	
\$t7	
\$s0	
\$s1	
\$s2	
\$s3	
\$s4	
\$s5	
\$s6	
\$s7	
\$t8	
\$t9	
\$k0	
\$k1	
\$gp	
\$sp	
\$ra	
pc	
hi	
lo	