Instructors:	Hacettepe University   Comp	tors: Prof. Dr. Prof. Dr. I seyin Tem Date: 09.04	Suleyman Mehmet K uçin 4.2019  5 20	Total	
Instructors:	Instruction   Associated   Instruction   Associated   Instruction   In	tors: Prof. Dr. Prof. Dr. I seyin Tem Date: 09.04	Suleyman Mehmet K uçin 4.2019  5 20	Total	
Assoc. Prof. Dr. Suleyman TOSUN Assist. Prof. Dr. Mehmet KOSEOGLU Dr. Hüseyin Temuçin Exam Date: 09.04.2019  Questions  1 2 3 4 5 Total  Marks 20 20 20 20 20 100  Earned  Q1. Convert the given MIPS program into the C code (using for loop).  MIPS ASSEMBLY CODE  addi \$\$1, \$0, 1  add \$\$0, \$0, \$0  addi \$\$t0, \$0, 10  for: beq \$\$0, \$\$t0, done \$\$11 \$\$s1, \$\$s1, 1  addi \$\$0, \$\$0, \$\$0  j for	Assoc   Assist   Dr. Hi   Duration: 120 minutes   Exam	Prof. Dr. Prof. Dr. Prof. Dr. Prof. Dr. Maseyin Tem Date: 09.04	Mehmet K uçin 4.2019 5 20	Total	
Assist. Prof. Dr. Mehmet KOSEOGLU   Dr. Hüseyin Temuçin	Assist Dr. Hi	Prof. Dr. 1 seyin Tem Date: 09.04	Mehmet K uçin 4.2019 5 20	Total	
Dr. Hüseyin Temuçin	Dr. History   Dr. History	seyin Tem Date: 09.04 4 20	uçin 4.2019 5 20	Total	
Duration: 120 minutes         Exam Date: 09.04.2019           Questions         1         2         3         4         5         Total           Marks         20         20         20         20         20         100           Earned         0         0         0         0           Q1. Convert the given MIPS program into the C code (using for loop).           MIPS ASSEMBLY CODE         addi \$\$1, \$0, 1         1           addi \$\$1, \$0, 1         1         4           addi \$\$1, \$0, \$0, \$0         2         4           addi \$\$1, \$0, \$0, \$0         3         4         5         Total           MIPS ASSEMBLY CODE         4         4         5         Total           MIPS ASSEMBLY CODE         4         4         5         Total           MIPS ASSEMBLY CODE         4         5         4         5         100           addi \$\$1, \$0, \$0, \$0         4         5         4         5         100           beq \$\$0, \$\$0, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0         50, \$0, \$0	Questions         1         2         3           Marks         20         20         20           Earned         Q1. Convert the given MIPS program into the C code (NIPS ASSEMBLY CODE addi \$s1, \$0, 1 addi \$s0, \$0, \$0 addi \$t0, \$0, 10 addi \$t0, \$0, 10 for: beq \$s0, \$t0, done s11 \$s1, \$s1, 1 addi \$s0, \$s0, 1 j for	Date: 09.04  4  20	5 20		
Marks 20 20 20 20 20 100  Earned 20 20 20 20 100  Q1. Convert the given MIPS program into the C code (using for loop).  MIPS ASSEMBLY CODE  addi \$s1, \$0, 1  add \$s0, \$0, \$0  addi \$t0, \$0, 10  for: beq \$s0, \$t0, done  sll \$s1, \$s1, 1  addi \$s0, \$s0, 1  j for	Marks  Earned  20 20 20 20  Earned  Q1. Convert the given MIPS program into the C code (  MIPS ASSEMBLY CODE  addi \$s1, \$0, 1  add \$s0, \$0, \$0  addi \$t0, \$0, 10  for: beq \$s0, \$t0, done  sll \$s1, \$s1, 1  addi \$s0, \$s0, 1  j for	20	20		
Marks 20 20 20 20 20 100  Earned 20 20 20 20 100  Q1. Convert the given MIPS program into the C code (using for loop).  MIPS ASSEMBLY CODE  addi \$s1, \$0, 1  add \$s0, \$0, \$0  addi \$t0, \$0, 10  for: beq \$s0, \$t0, done  sll \$s1, \$s1, 1  addi \$s0, \$s0, 1  j for	Marks  Earned  20 20 20 20  Earned  Q1. Convert the given MIPS program into the C code (  MIPS ASSEMBLY CODE  addi \$s1, \$0, 1  add \$s0, \$0, \$0  addi \$t0, \$0, 10  for: beq \$s0, \$t0, done  sll \$s1, \$s1, 1  addi \$s0, \$s0, 1  j for	20	20		
### Paragraph    ### Pa	### Parked  21. Convert the given MIPS program into the C code (1)  ### MIPS ASSEMBLY CODE    addi \$\$1, \$0, 1   add \$\$0, \$0, \$0   addi \$\$t0, \$0, 10  ### for: beq \$\$0, \$\$t0, done   \$\$11 \$\$1, \$\$1, \$   addi \$\$0, \$\$0, \$   j for			100	
addi \$s1, \$0, 1 add \$s0, \$0, \$0 addi \$t0, \$0, 10 for: beq \$s0, \$t0, done sll \$s1, \$s1, 1 addi \$s0, \$s0, 1 j for	Q1. Convert the given MIPS program into the C code (  MIPS ASSEMBLY CODE  addi \$s1, \$0, 1  add \$s0, \$0, \$0  addi \$t0, \$0, 10  for: beq \$s0, \$t0, done  sll \$s1, \$s1, 1  addi \$s0, \$s0, 1  j for	sing for	loop).		
MIPS ASSEMBLY CODE  addi \$s1, \$0, 1  add \$s0, \$0, \$0  addi \$t0, \$0, 10  for: beq \$s0, \$t0, done  sll \$s1, \$s1, 1  addi \$s0, \$s0, 1  j for	MIPS ASSEMBLY CODE  addi \$s1, \$0, 1  add \$s0, \$0, \$0  addi \$t0, \$0, 10  for: beq \$s0, \$t0, done  sll \$s1, \$s1, 1  addi \$s0, \$s0, 1  j for	sing for	loop).		
3	3				
C CODE					

**Q2** The C program given by (a) converts the uppercase letters of a character string to lowercase. Implement the MIPS program that provides the same functionality with given C program. Suppose that the character string is located at the memory addresses given by (b). Note that there is a character within each word field, and you must perform operation on least-significant byte.

C CODE	0x1C	'M'
<pre>int main() {</pre>	0x18	'B'
<pre>char* string = { 'H', 'E', 'L', 'L', 'O',</pre>	0x14	'B'
<pre>int length = 8;</pre>	0x10	<b>'</b> 0'
<pre>for(int i = 0; i &lt; length; i++) {     string[i] = string[i] + 32;</pre>	0x0C	'L'
}	0x08	'L'
	0x04	'E'
	0x00	'H'
(a)		(b)

## MIPS ASSEMBLY CODE

```
\# \$s0 = 32 , \$s1 = 8
```

**Q3.** Write machine code for the instruction given in bold. Indicate their instruction type, fill binary machine code by showing corresponding fields, and write their hexadecimal values into the boxes.

Address	Instruction	Opcode or funct	Register numbers		
0x90 fact:	addi \$sp, \$sp, -8		<del></del>		
0x94	sw \$a0, 4(\$sp)	#opcode: 0x2B	a0: 4, sp: 29		
0x98	sw \$ra, 0(\$sp)				
0x9C	addi \$t0, \$0, 2				
0xA0	slt \$t0, \$a0, \$t0				
0xA4	beq \$t0, \$0, else	#opcode: 0x04	t0: 8		
0xA8	addi \$v0, \$0, 1				
0xAC	addi \$sp, \$sp, 8				
0xB0	jr \$ra	#funct: 0x08	ra: 31		
0xB4 else:	addi \$a0, \$a0, -1				
0xB8	jal fact	#opcode: 0x03			
0xBC	lw \$ra, 0(\$sp)				
0xC0	lw \$a0, 4(\$sp)				
0xC4	addi \$sp, \$sp, 8				
0xC8	mul \$v0, \$a0, \$v0				
0xCC	jr \$ra				
Instruction	Type Binar	ry Code	Hex code	<b>)</b>	
sw \$a0, 4(\$	Ssp)				

Instruction	Type	Binary Code	Hex code
beq \$t0, \$0, else			

Instruction	Type	Binary Code	Hex code
jr \$ra			

Instruction	Type	Binary Code	Hex code
jal fact			

**Q4.** You are given a C program with a function call. Somebody converted the program into MIPS program. However, there are some mistakes in the MIPS code. Correct it so that it executes correctly.

C Code	MIPS Code
int main(){	main:
	addi \$sp, \$sp, -4
int f=2;	sw \$ra, 0(\$sp)
int g=3;	addi \$a0, \$0, 2
	addi \$a1, \$0, 3
y=sum(f,g);	j sum
return y;	addi \$s0, \$v0, \$0
}	lw \$ra, 0(\$sp)
int sum(int a, int b){	jr \$ra
return (a+b);}	
	sum:
	add \$v0, \$a0, \$a1
	jal \$ra

(a) (6 points) For 8-bit signed integer x = 0xA2, do the following calculations. Write your results in decimal.

$$x \ll 4 =$$

(b) (14 points) Below you will implement a logical and arithmetic shifter. The output, Y, will be the input, A shifted by 0 to 3 bits depending on the value of the 2-bit shift amount,  $shamt_{1:0}$ . For both shifters, when  $shamt_{1:0} = 00$ , Y = A.



## Arithmetic Right Shift

