Name-Last Name: _____ **Student ID:** _____

Section No (1,2,3?): ____

Hacettepe University	Computer Engineering Department
BBM234 Computer Organization	Instructors: Assoc. Prof. Dr. Suleyman TOSUN Assist. Prof. Dr. Mehmet KOSEOGLU Dr. Hüseyin Temuçin
1st Midterm Exam	
Duration: 120 minutes	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 $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
done:
```

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

```
int main() {
    char* string = {'H','E','L','L','O',
                    'B','B','M'};
    int length = 8;

    for(int i = 0; i < length; i++) {
       string[i] = string[i] + 32;
    }
}
```

(a)

0x1C	'M'
0x18	'B'
0x14	'B'
0x10	'0'
0x0C	'L'
0x08	'L'
0x04	'E'
0x00	'H'

(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		
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	Binary Code	Hex code
sw $a0, 4($sp)			

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

Q5.

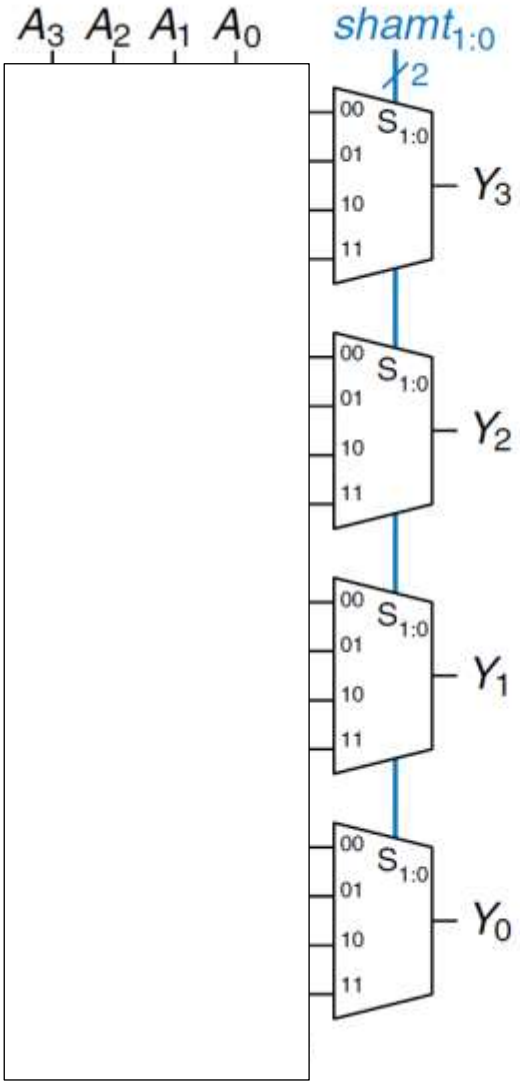
(a) (6 points) For 8-bit signed integer x = 0xA2, do the following calculations. Write your results in decimal.

x >> 4 = ☐ x << 4 = ☐

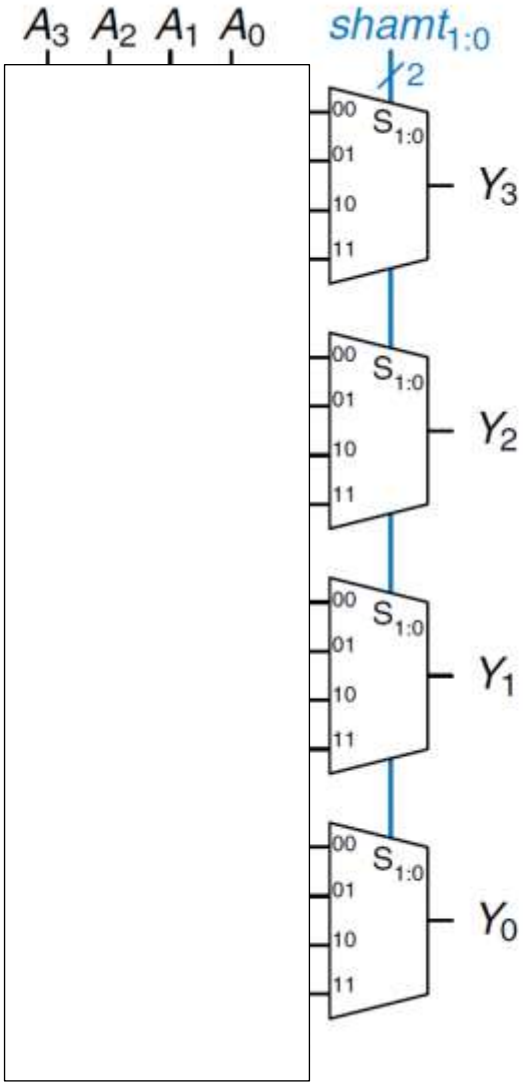
x >>>4= ☐

(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*.

Logical Right Shift



Arithmetic Right Shift



Good Luck ☺