Problem 1:

Write MIPS assembly program that reads 3 numbers from the memory, adds them, and stores the result back in the memory.

Solution:

```
.globl main
.data
Num1: .word 5
Num2: .word 3
Num3: .word 2
RES: .word 0

.text
main:
 la $t1, Num1
 la $t2, Num2
 la $t3, Num3
 la $t4, RES
 lw $s1, 0($t1)
 lw $s2, 0($t2)
 lw $s3, 0($t3)
 add $t5, $s1, $s2
 add $s4, $t5, $s3
 sw $s4, 0($t4)

li $v0,10
syscall
```

Another Solution:

```
.globl main
.data
Num1: .word 5
Num2: .word 3
Num3: .word 2
RES: .word 0
.text
main:
la $t1, Num1
lw $s1, 0($t1)
addi $t1,4
```

```
lw $s2, 0($t1)
addi $t1,4
lw $s3, 0($t1)
add $t5, $s1, $s2
add $s4, $t5, $s3
addi $t1,4
sw $s4, 0($t1)

li $v0,10
syscall
```

Problem 2:

Write MIPS assembly program that reads 2 numbers from the memory, finds the largest number and stores the largest in $s5.

Solution:

```
.globl main
.data
Num1: .word 20
Num2: .word 30
.text
main:
 la $t1, Num1
 lw $s1, 0($t1)
 la $t2, Num2
 lw $s2, 0($t2)
 beg  $s1, $s2, num1Bigger
 move $s5,$s2
 j exit
num1Bigger:
 move $s5,$s1
exit:
 li $v0,10
 syscall
```

Problem 3:

Write MIPS assembly program that sums up elements of an array that contains 8 elements then calculates the average and stores both the sum and average in the memory. Assume there is no div instruction.

Solution:

```
.globl main
.data
Arr: .word 3, 5, 1, 2, 10, 15, 4, 8
Sum: .word 0
Avg: .word 0

.text
main:
 la $t1, Arr

Loop:
 slti $t2, $s1, 8
 beq $t2, $zero, exit
 lw $s2,0($t1)
 add $s3, $s3, $s2
 addi $t1, 4
 addi $s1, 1
 j Loop

exit:
 sra $s4, $s3, 3
 sw $s3, 0($t1)
 addi $t1, 4
 sw $s4, 0($t1)

 li $v0,10
 syscall
```

Problem 4:

Write MIPS assembly program that after reading the elements of an array 0f 10 elements , gets the maximum and the minimum and stores them in registers $s5, $s6 respectively.

Hint: Assume that initial value of maximum =0 and minimum =1000.

Solution:

```
.globl main
.data
Arr: .word 3, 5, 1, 2, 10, 15, 4, 8, 7, 6

.text
main:
 la $t1, Arr
 li $s5 , 0
 li $s6, 1000
```

```
Loop:
 slti $t2, $s1, 10
 beq $t2, $zero, exit
 lw $s2,0($t1)
 blt $s2, $s6, min
 bgt $s2, $s5,max
 j continue

max:
 move $s5, $s2
 j continue

min:
 move $s6, $s2

continue:
 addi $t1, 4
 addi $s1, 1
 j Loop

exit:
 li $v0,10
 syscall
```

Problem 5:

Write MIPS assembly program that after reading the elements of an array of 10 elements that contains negative, positive and zero numbers, counts the zeros and stores it in $s7, and stores the negative numbers in another array.

Solution:

```
.globl main
.data
Arr: .word 3, -5, -1, 0, 10, 0, 4, -8, 7, 6
Arr2: .word 0 , 0, 0, 0, 0, 0, 0, 0, 0, 0

.text
main:
 la $t1, Arr
 la $t2, Arr2

Loop:
 slt $t3, $s1, 10
 beq $t3, $zero, exit
 lw $s2,0($t1)
 blt $s2, $zero, negative
 beq $s2, $zero, zeros
 j continue

negative:
 sw $s2,0($t2)
 addi $t2, 4
 j continue

zeros:
  addi $s7,1

continue:
 addi $t1, 4
 addi $s1, 1
 j Loop

exit:
 li $v0,10
 syscall
```

Problem 6:

Write MIPS assembly program that given a hexadecimal number 0xAB5F, should swap the values of bits from 0 to bits 7 with bits 8 to 15, so that the value become 0x5FAB using logic operations. Any value needed in the problem must be read from the memory.

Solution:

```
.globl main
.data
Value: .word 0xAB5F
MASK: .word 0x00FF
MASK2: .word 0xFF00
.text
main:
 la $t0, Value
 la $t1, MASK
 la $t2, MASK2

 lw $s0,0($t0)
 lw $s1,0($t1)
 lw $s2,0($t2)

 and $t3, $s0, $s1
 sll $t3, $t3,8

 and $t4, $s0, $s2
 srl $t4, $t4, 8

 or $s0, $t3, $t4
 sw $s0,0($t0)

 li $v0,10
 syscall
```