

**Problem 1:**

**Computer Science or Information Technology**

Instructor: Dr. G.E. Antoniou

Day, Month, Year

Day

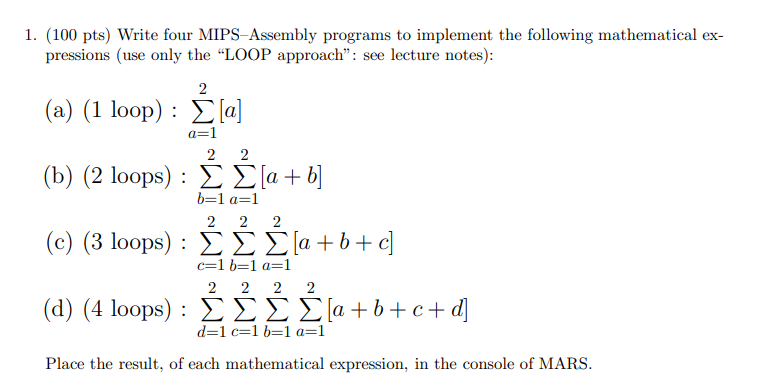
CSIT 502

Department of CSIT

Assessment

Module-7

Hidalgo, Rafael



**Solution (code)**

# 2

# SUM [a]

# a=1

.text # Informs the assembler that instructions follow

.globl main # Declare as global the label main

main:

li $t0, 1 # Loads 1 (a) into register $t0

loopa:

beq $t0, 3, exit # Branches if a "loop counter" $t0 is equal to 3

add $t1, $t1, $t0 # Adds the value stored in $t0 (a) with $t1 to $t1

addi $t0, $t0, 1 # Adds 1 to $t0 ("a" counter)

j loopa # Jumps to loopa

exit:

move $a0, $t1 # Moves the integer in $t1 to $a0

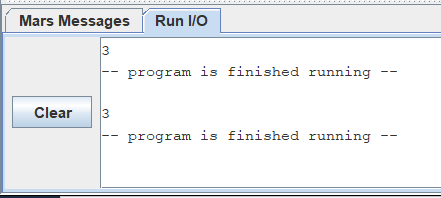
li $v0, 1 # System Call print Integer to Command line (code 1)

syscall

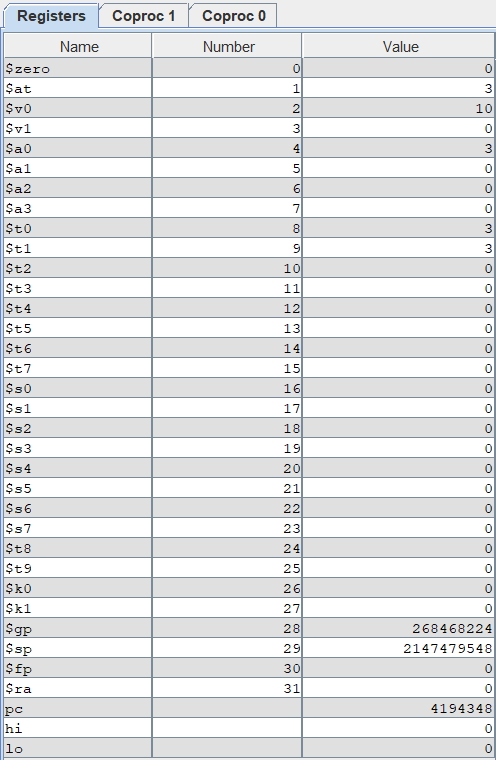
li $v0, 10 # System Call for Exit (code 10)

syscall

Sample Run (Console):



Sample Run (Registers area), only in decimal:



Result:

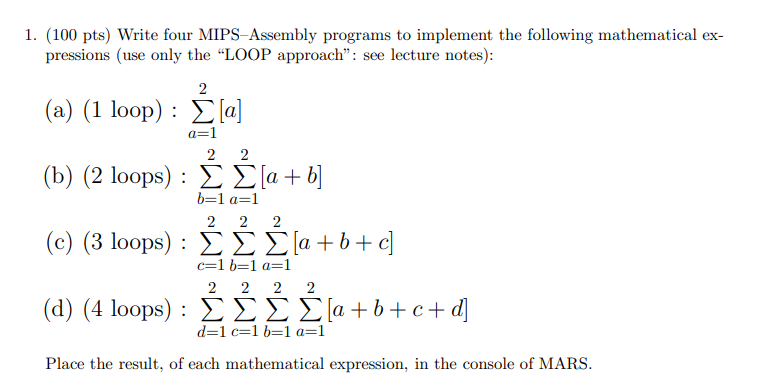
 = 3

This summation equation yields 3. As can be seen above, the appropriate result is displayed in the command window and in the registers area.

Brief Comments:

The program runs correctly, according to the specifications

**Problem 2:**



**Solution (code)**

**# 2 2**

**# SUM SUM [a + b]**

**# b=1 a=1**

**.text # Informs the assembler that instructions follow**

**.globl main # Declare as global the label main**

**main:**

**li $t0, 1 # Loads 1 (b) into register $t0**

**li $t1, 1 # Loads 1 (a) into register $t1**

**loopa:**

**beq $t1, 3, loopb # Branches if a "loop counter" $t1 is equal to 3**

**add $t3, $t3, $t1 # Adds the value stored in $t1 (a) with $t3 to $t3**

**add $t3, $t3, $t0 # Adds the value stored in $t0 (b) with $t3 to $t3**

**addi $t1, $t1, 1 # Adds 1 to $t1 ("a" counter)**

**j loopa # Jumps to loopa**

**loopb:**

**addi $t0, $t0, 1 # Adds 1 to $t0 ("b" counter)**

**beq $t0, 3, exit # Branches to exit if $t0 (b loop counter) equals 3**

**li $t1, 1 # Loads 1 into $t1**

**j loopa**

**exit:**

**move $a0, $t3 # Moves the integer in $t3 to $a0**

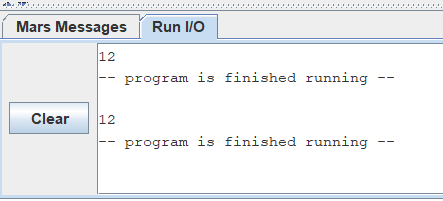
**li $v0, 1 # System Call print Integer to Cmd line (code 1)**

**syscall**

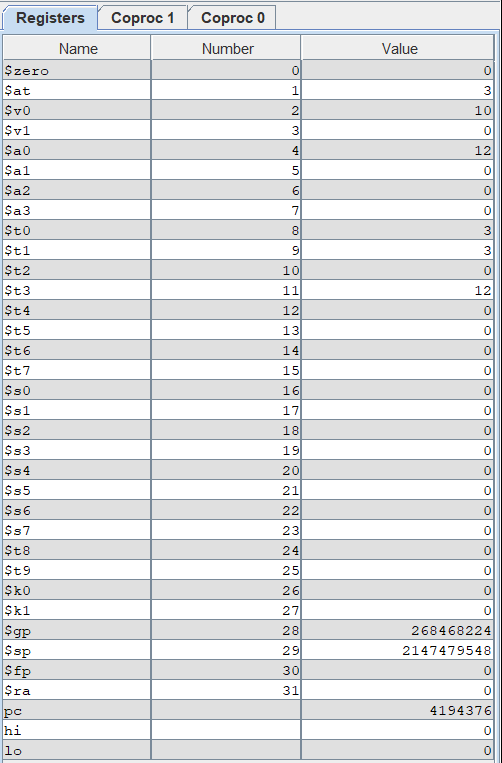
**li $v0, 10 # System Call for Exit (code 10)**

**syscall**

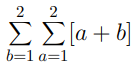
Sample Run (Console):



Sample Run (Registers area), only in decimal:



Result:

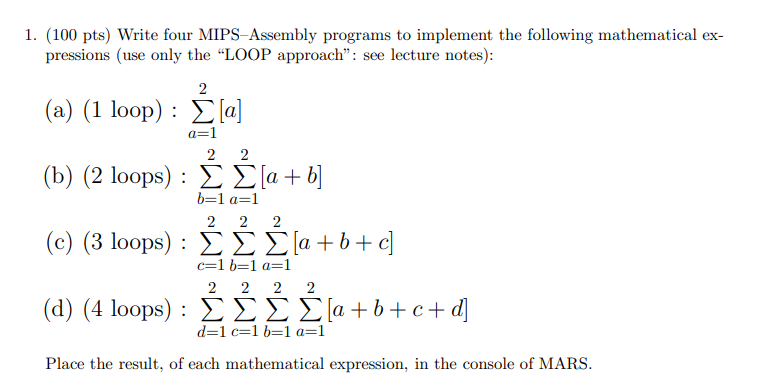
=12

This summation equation yields 12. As can be seen above, the appropriate result is displayed in the command window and in the registers area.

Brief Comments:

The program runs correctly, according to the specifications

**Problem 3:**



**Solution (code)**

# 2 2 2

# SUM SUM SUM [a + b + c]

# c=1 b=1 a=1

.text # Informs the assembler that instructions follow

.globl main # Declare as global the label main

main:

li $t0, 1 # Loads 1 (c) into register $t0

li $t1, 1 # Loads 1 (b) into register $t1

li $t2, 1 # Loads 1 (a) into register $t2

loopa:

beq $t2, 3, loopb # Branches if a "loop counter" $t2 is equal to 3

add $t3, $t3, $t2 # Adds the value stored in $t2 (a) with $t3 to $t3

add $t3, $t3, $t1 # Adds the value stored in $t1 (b) to $t3, adding to what's in $t3

add $t3, $t3, $t0 # Adds the value stored in $t0 (c) to $t3, adding to what's in $t3

addi $t2, $t2, 1 # Adds 1 to $t2 ("a" counter)

j loopa # Jumps to loopa

loopb:

addi $t1, $t1, 1 # Adds 1 to $t1 ("b" counter)

beq $t1, 3, loopc # Branches to loopc if $t1 (b loop counter) is equal to 3

li $t2, 1 # Loads 1 into $t2

j loopa

loopc:

addi $t0, $t0, 1 # Adds 1 to $t0 ("c" counter)

beq $t0, 3, exit # Branches to exit if $t0 (c loop counter) is equal to 3

li $t2, 1 # Loads 1 into $t2

li $t1, 1 # Loads 1 into $t1

j loopa

exit:

move $a0, $t3 # Moves the integer in $t3 to $a0

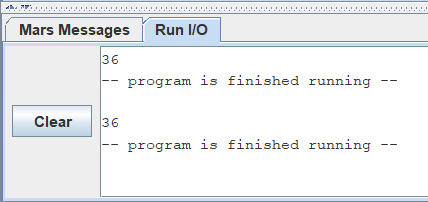
li $v0, 1 # System Call for printing Integer to Command line (code 1)

syscall

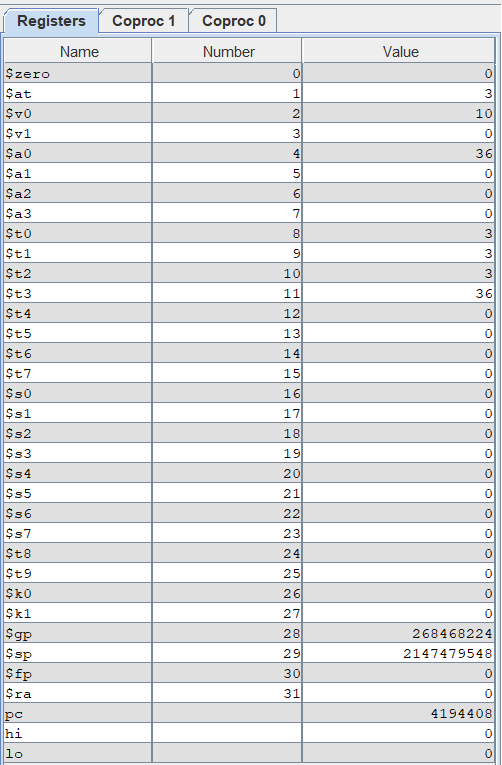
li $v0, 10 # System Call for Exit (code 10)

syscall

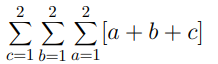
Sample Run (Console):



Sample Run (Registers area), only in decimal:



Result:

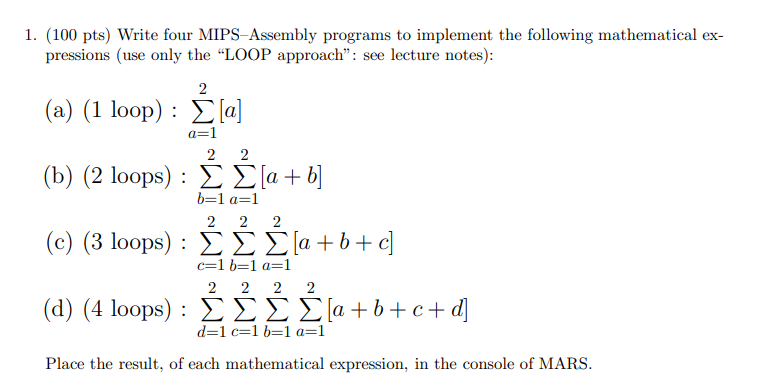
=36

This summation equation yields 36. As can be seen above, the appropriate result is displayed in the command window and in the registers area.

Brief Comments:

The program runs correctly, according to the specifications

**Problem 4:**



**Solution (code)**

# 2 2 2 2

# SUM SUM SUM SUM [a + b + c + d]

# d=1 c=1 b=1 a=1

.text # Informs the assembler that instructions follow

.globl main # Declare as global the label main

main:

li $t0, 1 # Loads 1 (d) into register $t0

li $t1, 1 # Loads 1 (c) into register $t1

li $t2, 1 # Loads 1 (b) into register $t2

li $t3, 1 # Loads 1 (a) into register $t3

loopa:

beq $t3, 3, loopb # Branches if a "loop counter" $t3 is equal to 3

add $t4, $t4, $t3 # Adds the value stored in $t3 (a) with $t4 to $t4

add $t4, $t4, $t2 # Adds the value stored in $t2 (b) to $t4, adding to what's in $t4

add $t4, $t4, $t1 # Adds the value stored in $t1 (c) to $t4, adding to what's in $t4

add $t4, $t4, $t0 # Adds the value stored in $t0 (d) to $t4, adding to what's in $t4

addi $t3, $t3, 1 # Adds 1 to $t3 ("a" counter)

j loopa # Jumps to loopa

loopb:

addi $t2, $t2, 1 # Adds 1 to $t2 ("b" counter)

beq $t2, 3, loopc # Branches to loopc if $t2 (b loop counter) is equal to 3

li $t3, 1 # Loads 1 into $t3

j loopa

loopc:

addi $t1, $t1, 1 # Adds 1 to $t1 ("c" counter)

beq $t1, 3, loopd # Branches to loopd if $t1 (c loop counter) is equal to 3

li $t3, 1 # Loads 1 into $t3

li $t2, 1 # Loads 1 into $t2

j loopa

loopd:

addi $t0, $t0, 1 # Adds 1 to $t0 ("d" counter)

beq $t0, 3, exit # Branches to exit if $t0 (d loop counter) is equal to 3

li $t3, 1 # Loads 1 into $t3

li $t2, 1 # Loads 1 into $t2

li $t1, 1 # Loads 1 into $t1

j loopa

exit:

move $a0, $t4 # Moves the integer in $t4 to $a0

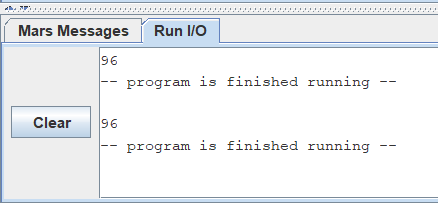
li $v0, 1 # System Call for printing Integer to Command line (code 1)

syscall

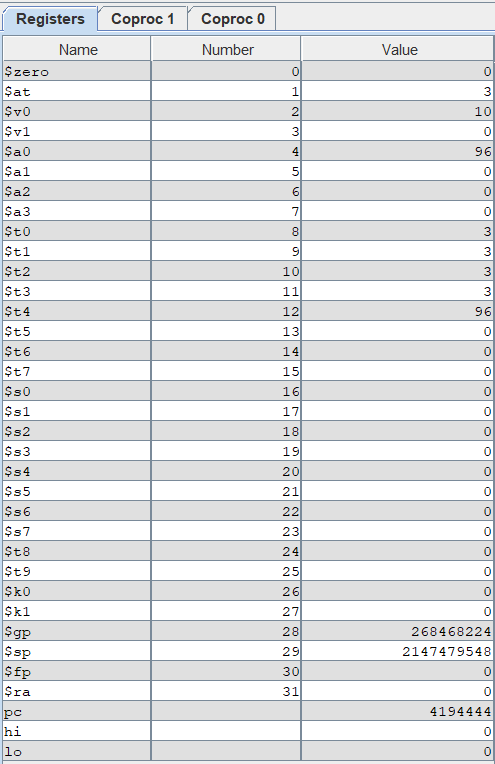
li $v0, 10 # System Call for Exit (code 10)

syscall

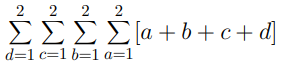
Sample Run (Console):



Sample Run (Registers area), only in decimal:



Result:

=96

This summation equation yields 96. As can be seen above, the appropriate result is displayed in the command window and in the registers area.

Brief Comments:

The program runs correctly, according to the specifications