e



**Problem 1:**

**Computer Science or Information Technology**

Instructor: Dr. G.E. Antoniou

Day, Month, Year

Day

CSIT 502

Department of CSIT

Assessment

Module-5

Hidalgo, Rafael

1. Write a MIPS Assembly program to ADD the numbers: 1, 2, 3, 4, 5, 6 (as in ex–1)

• Use ONLY the instructions: add and li

• The result of the addition (in decimal) should be in register: $t1

**Solution (code)**

.text

.globl main

main:

# EMULATES 1 + 2 + 3 + 4 + 5 + 6 USING add AND li

# Loads the values 1,2,3,4,5,6 into registers $t0,$t2,$t3,$t4,$t5,$t1 respectively.

li $t0,1

li $t2,2

li $t3,3

li $t4,4

li $t5,5

li $t1,6

# add 2 + 1 to equal 3 and loads it into $t0

add $t0,$t2,$t0

# add 3 + (2+1) to equal 6 and loads it into $t0

add $t0,$t3,$t0

# add 4 + (3 + (2+1)) to equal 10 and loads it into $t0

add $t0,$t4,$t0

# add 5 + (4 + (3 + (2+1))) to equal 15 and loads it into $t0

add $t0,$t5,$t0

# add 6 + (5 + (4 + (3 + (2+1)))) to equal 21 and loads it into $t1

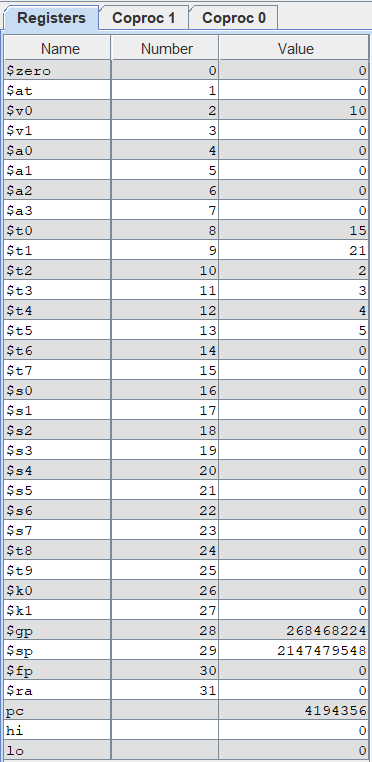
add $t1,$t1,$t0

# Exit (code 10) to exit the program

li $v0,10

syscall

Sample Run (Registers area), only in decimal:



Result:

6+5+4+3+2+1 = 21

As you can see above, register $t1 is equal to 21 using only add and li.

Brief Comments:

The program runs correctly, according to the specifications

**Problem 2:**

2. Write a MIPS Assembly program to ADD the numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9.

• Use ONLY the instructions: add and li

• Use any $t, $s registers

• The result of the addition (in decimal) should be in register: $t0

**Solution (code)**

.text

.globl main

main:

# EMULATES 1 + 2 + 3 + 4 + 5 + 6 USING add AND li

# Loads the values 1,2,3,4,5,6,7,8,9 into registers $t0,$t1,$t2,$t3,$t4,$t5,$t6,$t7,$s0 respectively.

li $t0,1

li $t1,2

li $t2,3

li $t3,4

li $t4,5

li $t5,6

li $t6,7

li $t7,8

li $s0,9

# add 2 + 1 to equal 3 and loads it into $t0

add $t0,$t1,$t0

# adds 3 plus current value of $t0 and loads it onto $t0

add $t0,$t2,$t0

# adds 4 plus current value of $t0 and loads it onto $t0

add $t0,$t3,$t0

# adds 5 plus current value of $t0 and loads it onto $t0

add $t0,$t4,$t0

# adds 6 plus current value of $t0 and loads it onto $t0

add $t0,$t5,$t0

# adds 7 plus current value of $t0 and loads it onto $t0

add $t0,$t6,$t0

# adds 8 plus current value of $t0 and loads it onto $t0

add $t0,$t7,$t0

# adds 9 plus current value of $t0 and loads it onto $t0

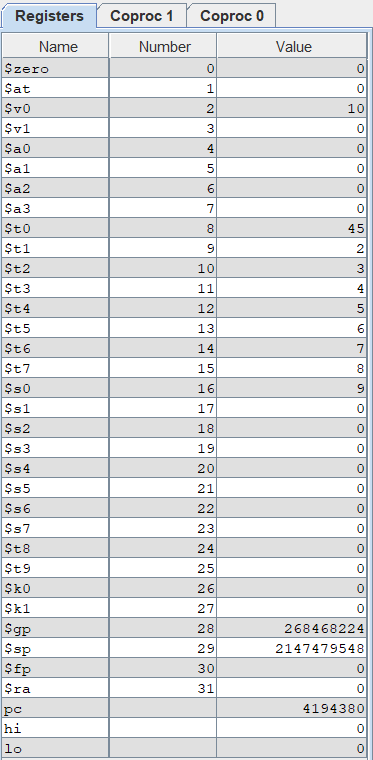
add $t0,$s0,$t0

# Exit (code 10) to exit the program

li $v0,10

syscall

Sample Run (Registers area), only in decimal:



Result:

9+8+7+6+5+4+3+2+1 = 45

As you can see above, register $t0 is equal to 45 using only add and li.

Brief Comments:

The program runs correctly, according to the specifications

**Problem 3:**

3. Write a MIPS Assembly program to ADD the numbers: 1, 2, 3, 4, 5, 6 (as in ex–2)

• Use ONLY the instructions: addi and li

• The result of the addition (in decimal) should be in register: $t1

**Solution (code)**

.text

.globl main

main:

# EMULATES 1+2+3+4+5+6 using only addi and li

# Loads the value 1 to $t0

li $t0, 1

# adds 2 to the value of $t0 and loads it to $t1

addi $t1, $t0, 2

# adds 3 to the value of $t1 and loads it to $t2

addi $t2, $t1, 3

# adds 4 to the value of $t2 and loads it to $t3

addi $t3, $t2, 4

# adds 5 to the value of $t3 and loads it to $t4

addi $t4, $t3, 5

# adds 6 to the value of $t4 and loads it to $t1

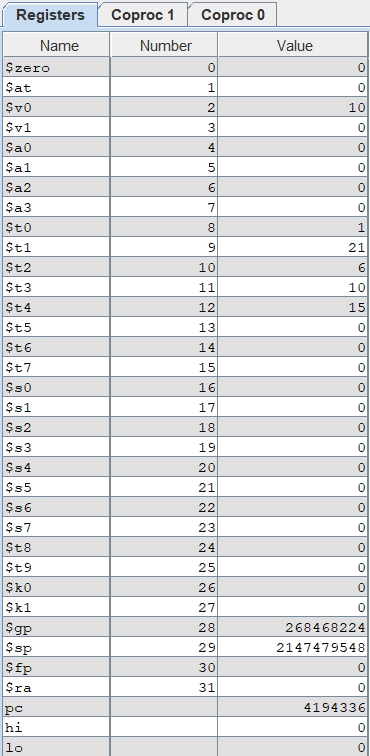
addi $t1, $t4, 6

# Exit (code 10) to exit the program

li $v0, 10

syscall

Sample Run (Registers area), only in decimal:



Result:

6+5+4+3+2+1 = 21

As you can see above, register $t1 is equal to 21 using only li and addi.

Brief Comments:

The program runs correctly, according to the specifications

**Problem 4:**

4. Write a MIPS Assembly program to ADD the numbers: 1, 2, 3, 4, 5, 6 (as in ex–3)

• Use ONLY the instructions: addi and li ... and ONLY the register: $t0

• The result of the addition (in decimal) should be in register: $t0

**Solution (code)**

.text

.globl main

main:

# EMULATES 1+2+3+4+5+6 using only addi and li and only register $t0

# Loads the value 1 to $t0

li $t0, 1

# adds 2 to the value of $t0 and loads it to $t0

addi $t0, $t0, 2

# adds 3 to the value of $t0 and loads it to $t0

addi $t0, $t0, 3

# adds 4 to the value of $t0 and loads it to $t0

addi $t0, $t0, 4

# adds 5 to the value of $t0 and loads it to $t0

addi $t0, $t0, 5

# adds 6 to the value of $t0 and loads it to $t0

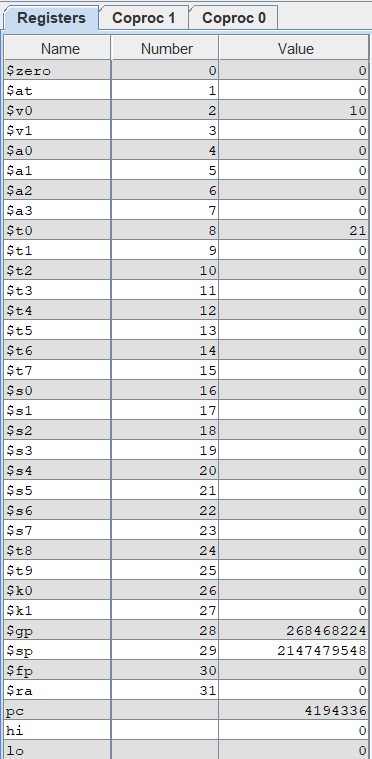
addi $t0, $t0, 6

# Exit (code 10) to exit the program

li $v0, 10

syscall

Sample Run (Registers area), only in decimal:



Result:

6+5+4+3+2+1 = 21

As you can see above, register $t0 is equal to 21 using only li and addi and register $t0.

Brief Comments:

The program runs correctly, according to the specifications