

**Problem 1:**

**Computer Science or Information Technology**

Instructor: Dr. G.E. Antoniou

Day, Month, Year

Day

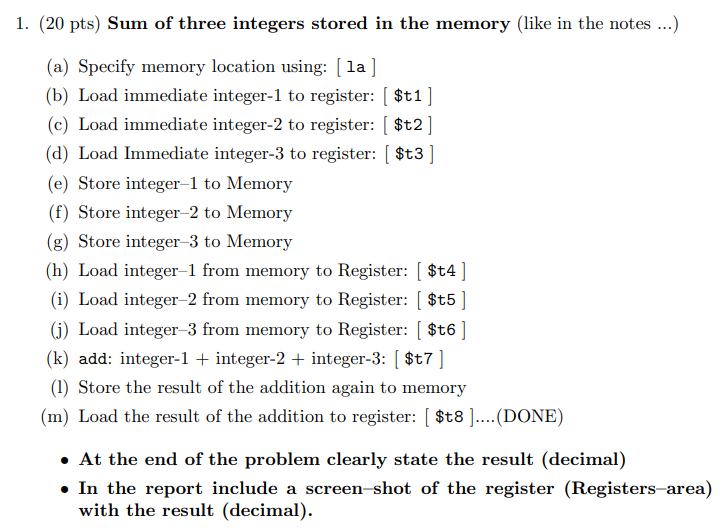
CSIT 502

Department of CSIT

Assessment

Module-8

Hidalgo, Rafael



**Solution (code)**

.data

Array: .space 16

.text

.globl main

main:

# Specify memory location using: [ la ]

la $t0, Array

# Load immediate integer-1 to register: [ $t1 ]

li $t1, 1

# Load immediate integer-2 to register: [ $t2 ]

li $t2, 2

# Load immediate integer-3 to register: [ $t3 ]

li $t3, 3

# Store integer–1 to Memory

sw $t1, 0($t0)

# Store integer–2 to Memory

sw $t2, 4($t0)

# Store integer–3 to Memory

sw $t3, 8($t0)

# Load integer–1 from memory to Register: [ $t4 ]

lw $t4, 0($t0)

# Load integer–2 from memory to Register: [ $t5 ]

lw $t5, 4($t0)

# Load integer–3 from memory to Register: [ $t6 ]

lw $t6, 8($t0)

#add: integer-1 + integer-2 + integer-3: [ $t7 ]

add $t7, $t4, $t5

add $t7, $t7, $t6

# Store the result of the addition again to memory

sw $t7, 12($t0)

# Load the result of the addition to register: [ $t8 ]....(DONE)

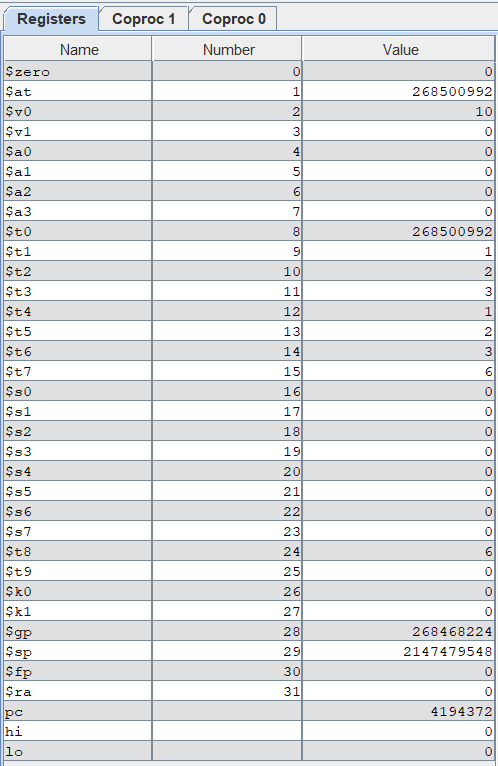
lw $t8, 12($t0)

# Exit

li $v0, 10

syscall

Sample Run (Registers area), only in decimal:



Result:

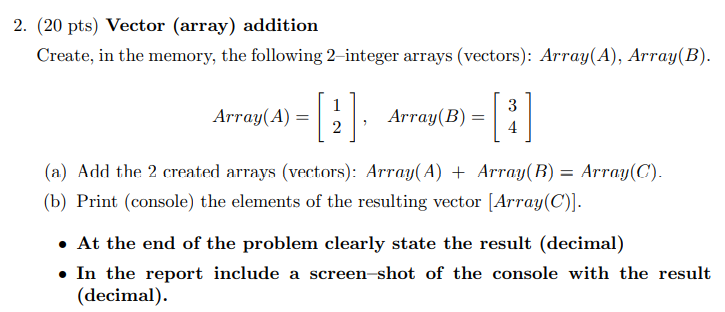
3+2+1 = 6

As can be seen above in the registers area, the appropriate result is displayed in $t8.

Brief Comments:

The program runs correctly, according to the specifications

**Problem 2:**



**Solution (code)**

**.text**

**.globl main**

**main:**

**# Specify memory location of ArrayA**

**la $t0, ArrayA**

**# Specify memory location of ArrayB**

**la $t1, ArrayB**

**# Specify memory location of Arrayc**

**la $t2, ArrayC**

**# Loads contents of ArrayA to registers**

**lw $t3, 0($t0)**

**lw $t4, 4($t0)**

**# Loads contents of ArrayB to registers**

**lw $t5, 0($t1)**

**lw $t6, 4($t1)**

**# Adds first element of each array together**

**add $t7, $t3, $t5**

**# Adds second element of each array together**

**add $t8, $t4, $t6**

**# Store the result of the first element addition to the first element of ArrayC**

**sw $t7, 0($t2)**

**# Store the result of the second element addition to the second element of ArrayC**

**sw $t8, 4($t2)**

**# prints on command console "The elements of ArrayC are:"**

**li $v0, 4**

**la $a0, prompt**

**syscall**

**# loads the result of the first element onto $a0**

**lw $a0, 0($t2)**

**li $v0, 1**

**syscall**

**# prints a newline**

**li $v0, 4**

**la $a0, skip**

**syscall**

**# loads the result of the second element onto $a0**

**li $v0, 1**

**lw $a0, 4($t2)**

**syscall**

**# exits**

**li $v0, 10**

**syscall**

**.data**

**ArrayA: .word 1, 2**

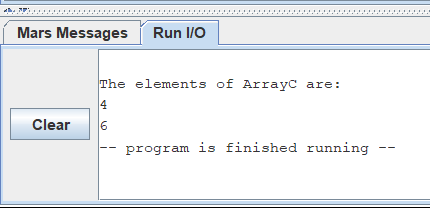
**ArrayB: .word 3, 4**

**ArrayC: .space 8**

**prompt: .asciiz "The elements of ArrayC are:\n"**

**skip: .asciiz "\n"**

Sample Run (Console):



Result:



+

= [ 4 ]

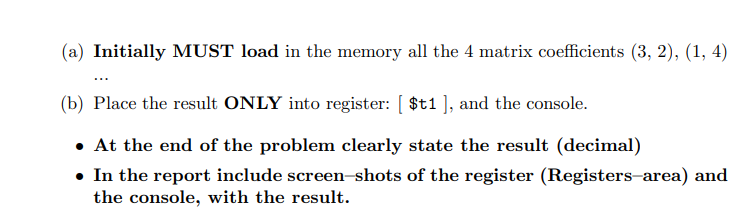
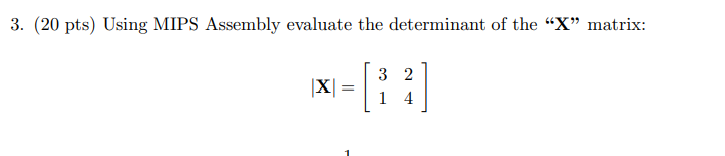
[ 6 ]

The vector addition yields the solution posted above. The command window also shows the same solution.

Brief Comments:

The program runs correctly, according to the specifications

**Problem 3:**



**Solution (code)**

.text

.globl main

main:

# Specify memory location of Matrix\_row1

la $t0, Matrix\_row1

# Specify memory location of Matrix\_row2

la $t2, Matrix\_row2

# Loads contents of Matrix\_row1 to registers

lw $t3, 0($t0)

lw $t4, 4($t0)

# Loads contents of Matrix\_row2 to registers

lw $t5, 0($t2)

lw $t6, 4($t2)

# multiplies 3 \* 4

mul $t7, $t3, $t6

# multiplies 2 \* 1

mul $t8, $t4, $t5

# performs (3 \* 4) - (2 \*1)

sub $t1, $t7, $t8

# prints on command console "The determinant is: "

li $v0, 4

la $a0, prompt

syscall

# loads the determinant to $a0

move $a0, $t1

li $v0, 1

syscall

# exits

li $v0, 10

syscall

.data

Matrix\_row1: .word 3, 2

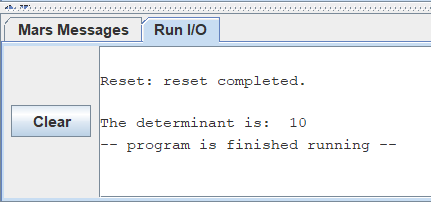
Matrix\_row2: .word 1, 4

ArrayC: .space 8

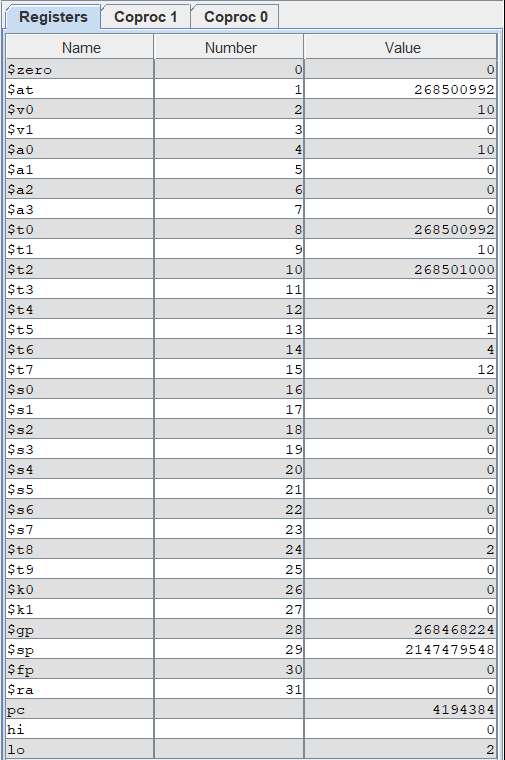
prompt: .asciiz "The determinant is: "

skip: .asciiz "\n"

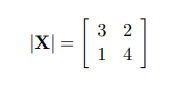
Sample Run (Console):



Sample Run (Registers area), only in decimal:



Result:

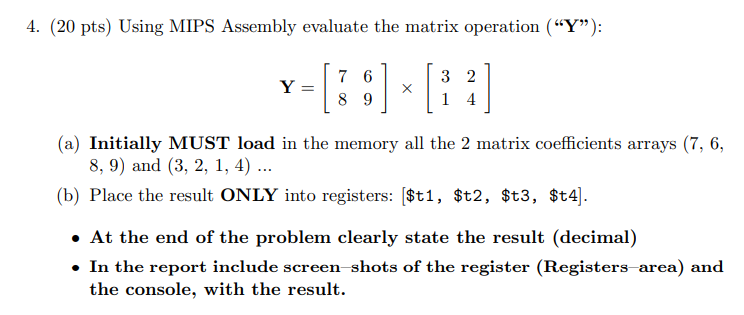
= 10

The determinant of the above matrix is 10 . 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)**

.text

.globl main

main:

# Specify memory location of Matrix\_1

la $s0, Matrix\_1

# Specify memory location of Matrix\_2

la $s1, Matrix\_2

# Loads contents of Matrix\_1 to registers

lw $s2, 0($s0)#7

lw $s3, 4($s0)#6

lw $s4, 8($s0)#8

lw $s5, 12($s0)#9

# Loads contents of Matrix\_2 to registers

lw $t5, 0($s1)#3

lw $t6, 4($s1)#2

lw $t7, 8($s1)#1

lw $t8, 12($s1)#4

# 7 x 3 + 6 x 1 = 27

mul $s6, $s2, $t5

mul $s7, $s3, $t7

add $t1, $s6, $s7

# 7 x 2 + 6 x 4 = 38

mul $s6, $s2, $t6

mul $s7, $s3, $t8

add $t2, $s6, $s7

# 8 x 3 + 9 x 1 = 33

mul $s6, $s4, $t5

mul $s7, $s5, $t7

add $t3, $s6, $s7

# 8 x 2 + 9 x 4 = 52

mul $s6, $s4, $t6

mul $s7, $s5, $t8

add $t4, $s6, $s7

# prints on command console "See the new array below: "

li $v0, 4

la $a0, prompt

syscall

# implements newline

li $v0, 4

la $a0, skip

syscall

# loads the first element of the first row of the array to console

move $a0, $t1

li $v0, 1

syscall

# implements space

li $v0, 4

la $a0, gap

syscall

# loads the second element of the first row of the array to console

move $a0, $t2

li $v0, 1

syscall

# implements newline

li $v0, 4

la $a0, skip

syscall

# loads the first element of the second row of the array to console

move $a0, $t3

li $v0, 1

syscall

# implements space

li $v0, 4

la $a0, gap

syscall

# loads the second element of the second row of the array to console

move $a0, $t4

li $v0, 1

syscall

# exits

li $v0, 10

syscall

.data

Matrix\_1: .word 7, 6, 8, 9

Matrix\_2: .word 3, 2, 1, 4

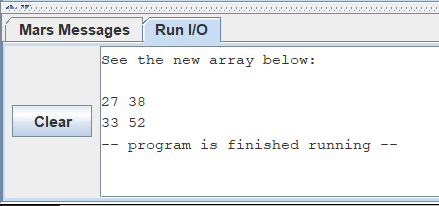
ArrayC: .space 8

prompt: .asciiz "See the new array below:\n "

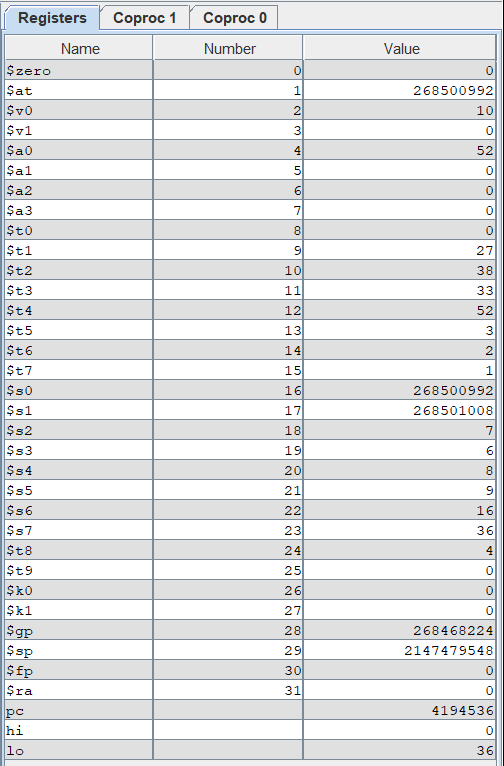
skip: .asciiz "\n"

gap: .asciiz " "

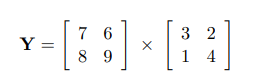
Sample Run (Console):



Sample Run (Registers area), only in decimal:



Result:



=

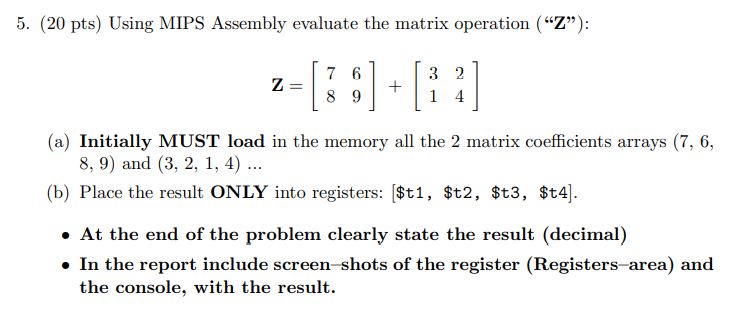
|  |  |
| --- | --- |
| 27 | 38 |
| 33 | 52 |

This array multiplication yields the above matrix. As can be seen above, the appropriate results are displayed in the command window and in the registers area.

Brief Comments:

The program runs correctly, according to the specifications

**Problem 5:**

**Solution (code)**

.text

.globl main

main:

# Specify memory location of Matrix\_1

la $s0, Matrix\_1

# Specify memory location of Matrix\_2

la $s1, Matrix\_2

# Loads contents of Matrix\_1 to registers

lw $s2, 0($s0)#7

lw $s3, 4($s0)#6

lw $s4, 8($s0)#8

lw $s5, 12($s0)#9

# Loads contents of Matrix\_2 to registers

lw $t5, 0($s1)#3

lw $t6, 4($s1)#2

lw $t7, 8($s1)#1

lw $t8, 12($s1)#4

# 7 + 3

add $t1, $s2, $t5

# 6 + 2

add $t2, $s3, $t6

# 8 + 1

add $t3, $s4, $t7

# 9 + 4

add $t4, $s5, $t8

# prints on command console "See the new array below: "

li $v0, 4

la $a0, prompt

syscall

# implements newline

li $v0, 4

la $a0, skip

syscall

# loads the first element of the first row of the array to console

move $a0, $t1

li $v0, 1

syscall

# implements space

li $v0, 4

la $a0, gap

syscall

# loads the second element of the first row of the array to console

move $a0, $t2

li $v0, 1

syscall

# implements newline

li $v0, 4

la $a0, skip

syscall

# loads the first element of the second row of the array to console

move $a0, $t3

li $v0, 1

syscall

# implements space

li $v0, 4

la $a0, gap

syscall

# loads the second element of the second row of the array to console

move $a0, $t4

li $v0, 1

syscall

# exits

li $v0, 10

syscall

.data

Matrix\_1: .word 7, 6, 8, 9

Matrix\_2: .word 3, 2, 1, 4

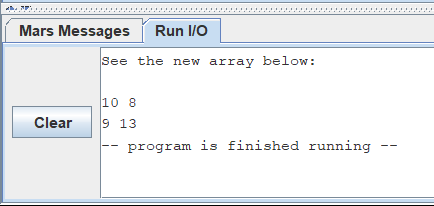
ArrayC: .space 8

prompt: .asciiz "See the new array below:\n "

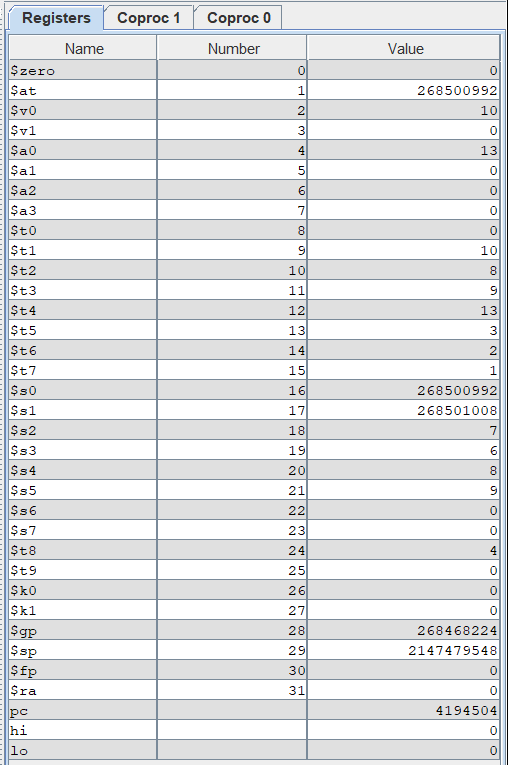
skip: .asciiz "\n"

gap: .asciiz " "

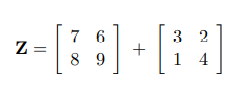
Sample Run (Console):



Sample Run (Registers area), only in decimal:



Result:



=

|  |  |
| --- | --- |
| 10 | 8 |
| 9 | 13 |

This array addition yields the above matrix. As can be seen above, the appropriate results are displayed in the command window and in the registers area.

Brief Comments:

The program runs correctly, according to the specifications