Assembly language-I (Arithmetic operations)

Paul Collado

Department of Computer Science

Montclair State University

CMPT280

Assembly Language

11/11/2014

Exercise 1:

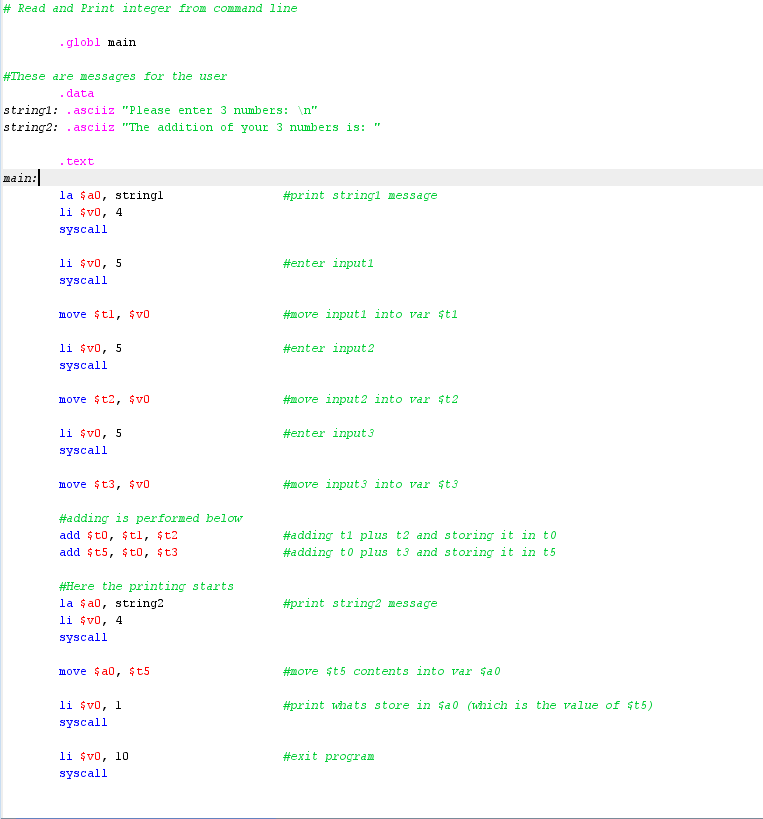
1. Problem:

Read and output in the command line MARS windows

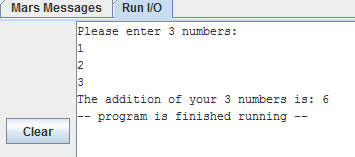
1. Read three integers from the command line window
2. Add the three integers
3. Print the result in the command line window
4. Solution:
5. Analysis:

I approach this program by creating 2 string to hold messages that user will see, enter code to read user’s input and then moving it to a $t variable, then I add $t1 and $t2 and store the result in a new $t0variable, then I add $t0 with the last $t3 variable and store in a final $t5 variable. Finally I called the system call to print the message and the result

1. Program code:



1. Example:



Exercise 2

1. Problem:

Print-out the result, in the command line (window), for the following expression:

($s0 - $s1 - $s2) + $s3

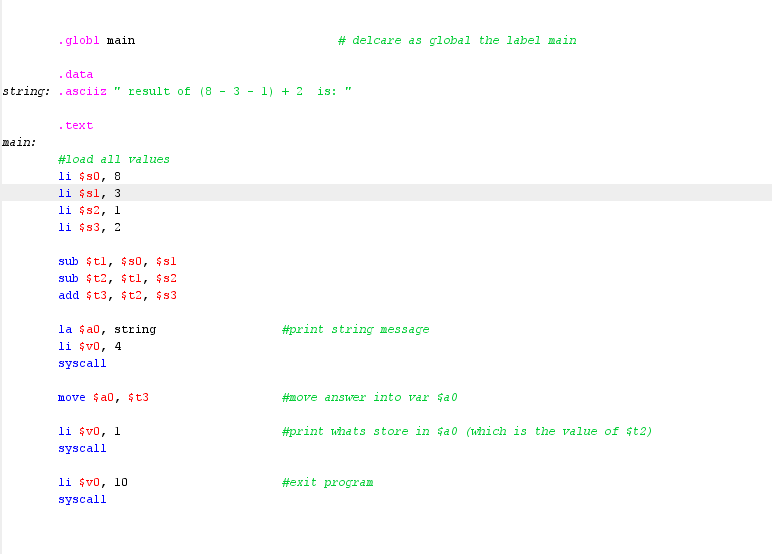
Where,

* $s0 = 8
* $s1 = 3
* $s2 = 1
* $s3 =2

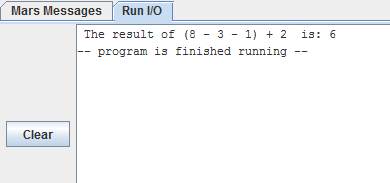
1. Solution:
2. Analysis:

For this problem I stored all numbers into respective variables and then subtracted them in pairs of two and then just add it with last variable. Finally I just added a message to make the program look nicer

1. Program code:



1. Example:



Exercise 3

1. Problem:

Place the result, into register $t0, for the following equation:

$t0 = $s0/8 – 2 \* $s1 + $s2

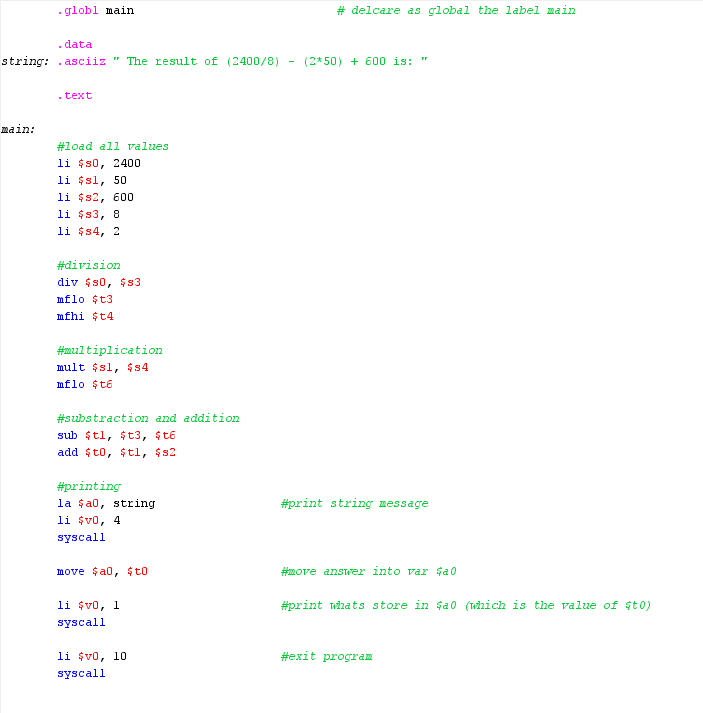
Where

* $s0 = 2400
* $s1 = 50
* $s2 = 600

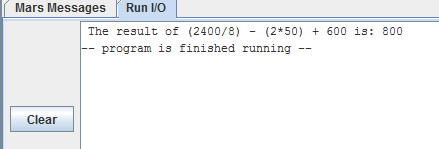
1. Solution
2. Analysis:

For this problem I decided to store all values on variables, then I first I divided 2 variables and stored the low in a new variable $t3, then I multiplied the other 2 variables and store the low in a new variable $t6, then I sub $t6 - $t3 and store it in a new variable $t1 and finally add $t1 with the last variable $s2 and store the final answer in $t0 variable

1. Program code



1. Example:



Exercise 4

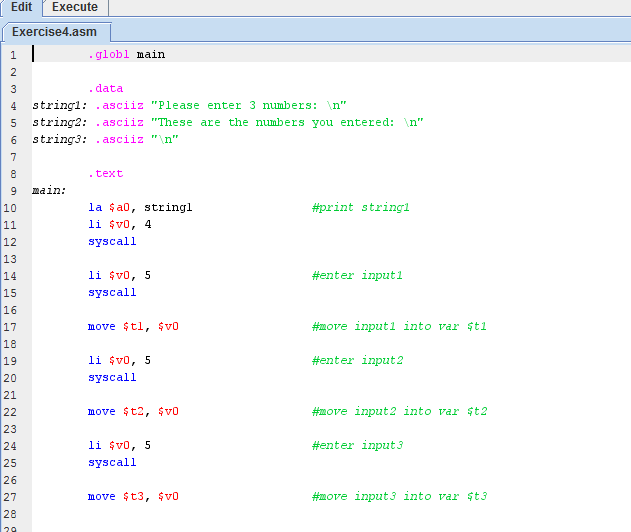
1. Problem:

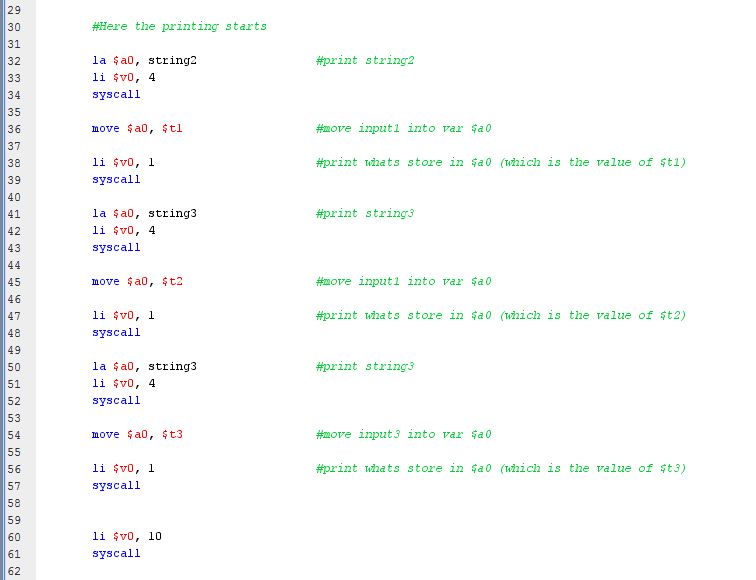
Create your own arithmetic example and code it in Assembly language

1. Solution:
2. Analysis:

For this exercise I decided to create a program that read three different inputs from the command prompt and then it prints the three of them. The challenge here was that I had to move every entered value from $a0 to a variable and then move it back when I needed to print it

1. Program code:





1. Example:

