Lab 3: MIPS General-Purpose Registers

Objectives:

After completing this lab, you will:

- Understand MIPS 32 * 32-bit registers
- Write simple MIPS programs using MIPS registers
- Convert simple C code into MIPS code

MIPS Registers

- MIPS architecture provides for 32 special locations, built directly into the CPU, each of them able to store 32 bits of information (1 word), called "registers".
- A small number of registers that can be accessed easily and quickly will allow the CPU to execute instructions very fast.

Exercise 1: Adding two operands

Use the following code to convert the addition operation of C code into MIPS code and print the result on the output screen.

C code : f=(g + h)

Exercise1.asm

```
1   .data
2   msg: .asciiz "The result of addition is: "
3
4   .text
5
6   li $t0,9
7   li $t1,5
8
9   Add $t3,$t0,$t1
10
11  li $v0,4
12  la $a0,msg
13  syscall
14
15  li $v0,1
16  move $a0,$t3
17  syscall
18
19  li $v0,10
20  syscall
```

Exercise 2: Modify the above program to compile the following C code into MIPS code:

C code: f=(g + h) – (i + j);

Edit | Execute

Exercise2.asm | Exercise11.asm | mips4.asm*

```
1   .data
2   msg: .asciiz "The result of the operation is: "
3
4   .text
5
6   li $t0,9
7   li $t1,5
8   li $t2,6
9   li $t3,2
10
11  add $t4,$t0,$t1
12  add $t5,$t2,$t3
13  sub $t6,$t4,$t5
14
15  li $v0,4
16  la $a0,msg
17  syscall
18
19  li $v0,1
20  move $a0,$t6
21  syscall
22
23  li $v0,10
24  syscall
```

Exercise 3: Modify the previous program to compile the following C code into MIPS code using 4 variables as follow:

g: .word 9

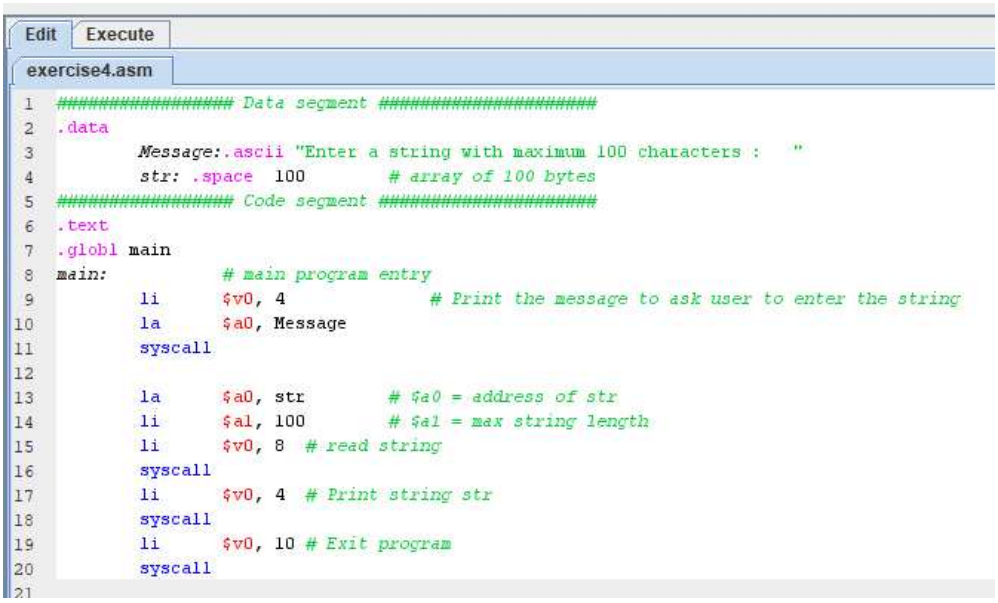
h: .word 5

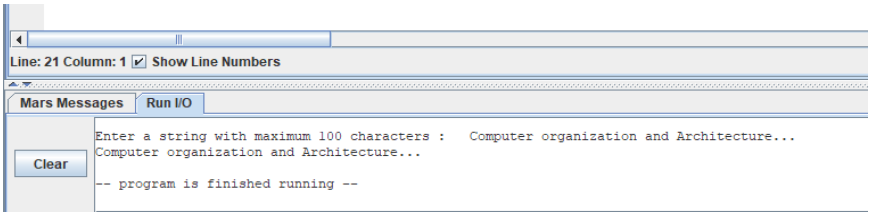
i: .word 6

j: .word 2

Exercise 4:

Write a program to ask the user to enter a string with maximum 100 characters and then print it.





Exercise 5:

Write a program to ask the user to enter two integers A and B and then display the result of computing the expression: A + 2B - 5.