**Programming with MARS IDE  
Console Commands**

The purpose of this lab is to introduce you to the layout and structure of the Mars IDE development tool in addition to Array programming and console commands. In this lab, Fibonacci numbers are generated and printed in the console of the MIPS environment.

**Introduction:**

By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two.

The Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181

**Procedure:**

I. Follow the same procedure in Lab1.

II. From the main menu, choose “File”🡪 “New”.

III. Write the code provided in the screen shot below.

***The code:***

# ----------------------------Lab 5-------------------------------------

#-----------------------Generate Fibonacci numbers----------------  
#-----------------------Print on the MIPS console-------------------

#-------------------------By: Dr. Abdallah---------------------------

# Compute several Fibonacci numbers and put in array, then print

.data

fibs:.word 0 : 19 # "array" of words to contain fib values

size: .word 19 # size of "array" (agrees with array declaration)

.text

la $s0, fibs # load address of array

la $s5, size # load address of size variable

lw $s5, 0($s5) # load array size

li $s2, 1 # 1 is the known value of first and second Fib. number

sw $s2, 0($s0) # F[0] = 1

sw $s2, 4($s0) # F[1] = F[0] = 1

addi $s1, $s5, -2 # Counter for loop, will execute (size-2) times

# Loop to compute each Fibonacci number using the previous two Fib. numbers.

loop: lw $s3, 0($s0) # Get value from array F[n-2]

lw $s4, 4($s0) # Get value from array F[n-1]

add $s2, $s3, $s4 # F[n] = F[n-1] + F[n-2]

sw $s2, 8($s0) # Store newly computed F[n] in array

addi $s0, $s0, 4 # increment address to now-known Fib. number storage

addi $s1, $s1, -1 # decrement loop counter

bgtz $s1, loop # repeat while not finished

# Fibonacci numbers are computed and stored in array. Print them.

la $a0, fibs # first argument for print (array)

add $a1, $zero, $s5 # second argument for print (size)

jal print # call print routine.

# The program is finished. Exit.

li $v0, 10 # system call for exit

syscall # Exit!

###############################################################

# Subroutine to print the numbers on one line.

.data

space:.asciiz " " # space to insert between numbers

head: .asciiz "The Fibonacci numbers are:\n"

.text

print:add $t0, $zero, $a0 # starting address of array of data to be printed

add $t1, $zero, $a1 # initialize loop counter to array size

la $a0, head # load address of the print heading string

li $v0, 4 # specify Print String service

syscall # print the heading string

out: lw $a0, 0($t0) # load the integer to be printed (the current Fib. number)

li $v0, 1 # specify Print Integer service

syscall # print fibonacci number

la $a0, space # load address of spacer for syscall

li $v0, 4 # specify Print String service

syscall # print the spacer string

addi $t0, $t0, 4 # increment address of data to be printed

addi $t1, $t1, -1 # decrement loop counter

bgtz $t1, out # repeat while not finished

jr $ra # return from subroutine

# End of subroutine to print the numbers on one line

###############################################################

* ***Observe*** the registers/Memory locations values after each Step Run.

Questions:

1. What does the final value of $s2 represent?
2. What does the final value of $s3 represent?
3. What does the final value of $s4 represent?
4. What does the final value of $s5 represent?
5. Modify the code to generate 21 Fibonacci numbers

**Things to turn in as your Lab 5 Report, attached in this order:**

1- Your name, Course Number, Lab Number and Date

2- Screen shot of the Program

3- Results/Observations such as the final screen shot of the registers and memory locations

4- Questions with answers

6- References if any