TO ADD 10 no’s:

#this program computes sum of first 10 natural numbers and stores the sum=55 in hexadecimal form (0x37) in the register 9

.data

.text

main:

li $t0, 0 #setting count = 0;

li $t1, 0 #setting sum = 1;

while: #This is a branch if greater than.

bgt $t0, 10, exit

add $t1, $t1, $t0 #sum = sum + count

addi $t0, $t0, 1 #count = count + 1

j while # this will keep repeating.

exit:

li $v0,10 # exit

TO PRINT “Hello MIPS”:

.data

msg:.asciiz "Hello MIPS"

.text

li $v0,4 la $a0,msg

syscall

TO REVERSE INPUT STRING (“ABC” - “CBA”):

.data

input: .space 256

output: .space 256

.text

.globl main

main:

li $v0, 8 # Ask the user for the string they want to reverse

la $a0, input # storing it in 'input'

li $a1, 256 # Only 256 chars/bytes allowed

syscall

li $v0, 4 # output the string to verify

la $a0, input

syscall

jal strlen # JAL to strlen function, saves return address to $ra

add $t1, $zero, $v0 # Copy some of our parameters for our reverse function

add $t2, $zero, $a0 # We need to save our input string to $t2, it gets

add $a0, $zero, $v0 # butchered by the syscall.

li $v0, 1 # This prints the length that we found in 'strlen'

syscall

reverse:

li $t0, 0 # Set t0 to zero to be sure

li $t3, 0 # and the same for t3

reverse\_loop:

add $t3, $t2, $t0 # $t2 is the base address for our 'input' array, add loop index

lb $t4, 0($t3) # load a byte at a time according to counter

beqz $t4, exit # We found the null-byte

sb $t4, output($t1) # Overwrite this byte address in memory

subi $t1, $t1, 1 # Subtract our overall string length by 1 (j--)

addi $t0, $t0, 1 # Advance our counter (i++)

j reverse\_loop # Loop until we reach our condition

exit:

li $v0, 4 # Print

la $a0, output # the string

syscall

li $v0, 10 # exit()

syscall

# strlen:

# a0 is our input string

# v0 returns the length

# -- This function loops over the character array until it encounters

# the null byte, interestingly, the 0x0a character is stored by default

# for input strings requested through the syscall. So we just subtract one

# from the end result.

strlen:

li $t0, 0

li $t2, 0

strlen\_loop:

add $t2, $a0, $t0

lb $t1, 0($t2)

beqz $t1, strlen\_exit

addiu $t0, $t0, 1

j strlen\_loop

strlen\_exit:

subi $t0, $t0, 1

add $v0, $zero, $t0

add $t0, $zero, $zero

jr $ra