CSE341 PROJECT REPORT

Name:Saugat Bikram Khadka
UB Person #: 50156469
Date: 11/30/17

1. CODE
CALCULATOR

```
.data
firstnum: .asciiz"\n\nplease input the first num: "
secondnum: .asciiz"input the second num: "
operation: .asciiz"input the operation: "
result: .asciiz"the result is: "
wrong_operation: .asciiz "cant perform any action."
.text
main:

start:
li $v0, 4              #to print the first input message
la $a0, firstnum
syscall

li $v0, 5
syscall
move $t0, $v0    #saving the first    number to t0

li $v0, 4
la $a0, secondnum    #printing second input message
syscall


li $v0, 5
syscall
move $t1, $v0    #saving the second number to t1

li $v0, 4
la $a0, operation    # printing the operation message
syscall


li $v0, 5
syscall
move $t2, $v0

li $v0, 4
la $a0, result    #printing the result message
```

```
syscall


beq $t2,1 ,addition #if user enters 1 do addition
beq $t2,2, subtraction      #if user enters 2    do subtraction
beq $t2,3, multiplication    # if user enters 3 do multiplication
beq $t2,4, division    #if user enters 4 do division
bgt $t2,4,    exit #if user enters more than    4 do nothing

addition: add $a0, $t0, $t1
li $v0, 1
syscall
j start


subtraction: sub $a0, $t0, $t1
li $v0, 1
syscall
j start


  multiplication: mul $a0, $t0, $t1
li $v0, 1
syscall
j start

division: div $a0, $t0, $t1
li $v0, 1
syscall
j start      # jump back    to start to do the operation again

exit:
li $v0, 4
la $a0, wrong_operation
syscall
j start
```

CODE FOR HAMMING DISTANCE

```
.data

input1: .asciiz "\n\nplease enter the first string:\n"
input2: .asciiz "\nplease enter the second string:\n"
answer: .asciiz "\nthe hamming distance is:"

string1: .space 3      #3 bytes to hold string of length two plus a
terminating charcter
string2: .space 3

.text
main:
li $v0, 4                        #to print the first string message
la $a0, input1
syscall

li $v0, 8                        #ready to print a string
la $a0, string1
li $a1,3
move $s0, $a0      #saving the string into $s0 register
syscall


li $v0, 4                    # to print the second string message
la $a0, input2
syscall

li $v0, 8                    #ready to print another string
la $a0, string2
li $a1,3
move $s1, $a0      # saving the string into $s1 register
syscall
li $t1,0    #counter = 0

while:
```

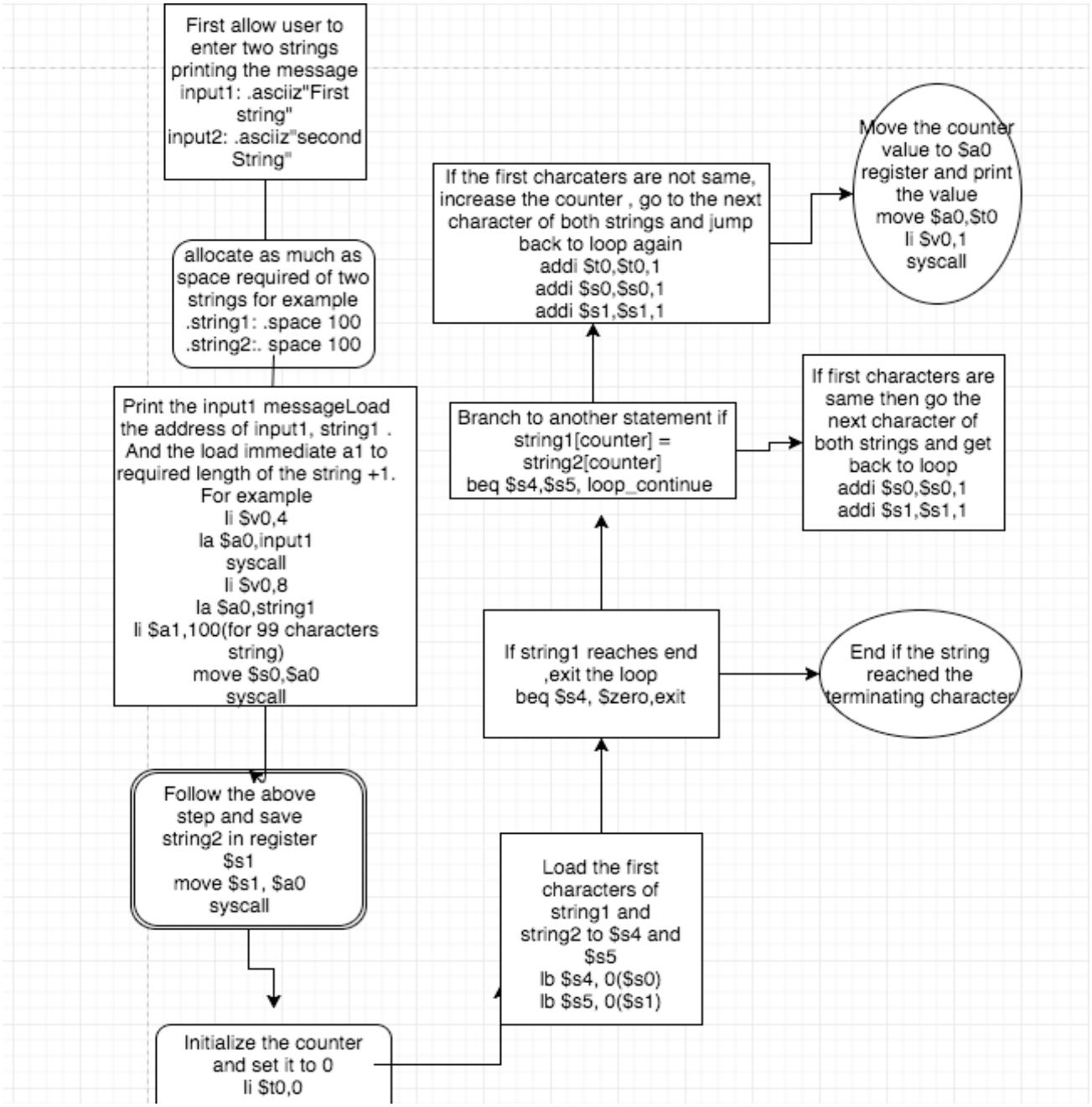
```
lb $s4, 0($s0)    #loading the first character of string1 to $s4
lb $s5, 0($s1)     #loading the second character of string1 to $s5

beq $s4, $zero exit          # if string 1 hits the terminating character exit

beq $s4,$s5, loop_continue      #branch to loop_continue if the first
characters are same
addi $t1, $t1, 1          # increases the counter if the charcaters are
different
addi $s0,$s0,1      # next charcater of string1
addi $s1, $s1,1      #next character of string2
j while        #jumps back to the while

loop_continue:      # if the first characters are same
addi $s0,$s0,1    # second character of string1
addi $s1,$s1,1      #second charcater of string2
j while      #jumb back to the while


exit:      #if it hits the null character end the loop


li $v0,4      # ready to print the answer message
la $a0, answer
syscall


move $a0,$t1    #save the value of counter to $a0
li $v0,1      #print the value
syscall
j main      # go back to the main again to start the program again

li $v0,10    #end of program
syscall
```

CODE FOR EUCLIDEAN DISTANCE

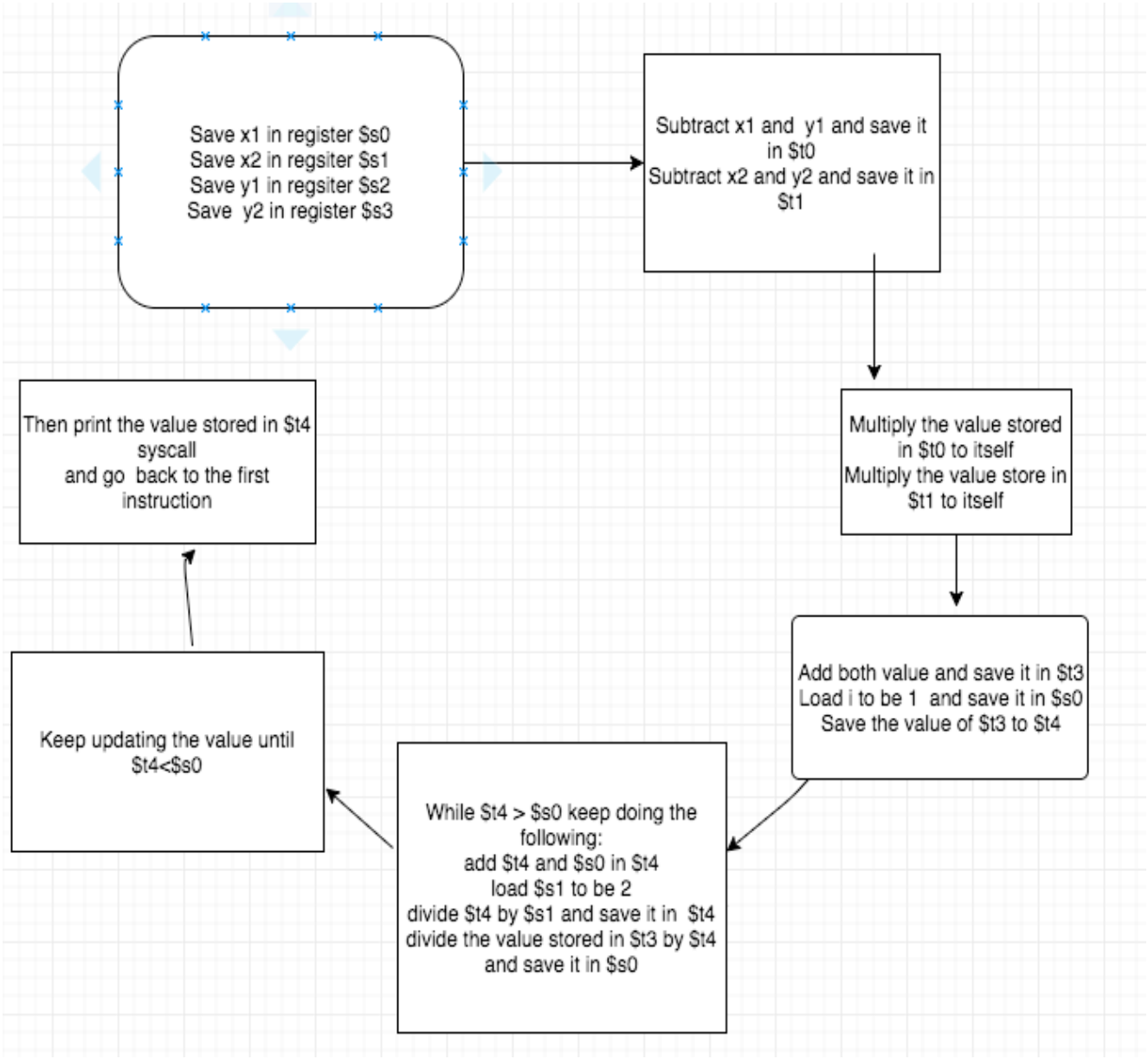
```
.data

TwoD: .asciiz "\n\nEuclidean distance for 2D\n"
x1: .asciiz"Enter the x1 value: "
x2: .asciiz"Enter the x2 value: "
y1: .asciiz"Enter the y1 value: "
y2: .asciiz"Enter the y2 value: "
newdistance: .asciiz"The euclidean distance for the above values is: "
.text

main:
distance2D:
li $v0, 4
la $a0, TwoD
syscall

li $v0,4
la $a0, x1
syscall

li $v0,5        #user to input the x1 number and save it in $s0
syscall
move $s0, $v0

li $v0,4
la $a0, x2
syscall

li $v0,5      #user to input the x2 number and save it in $s1
syscall
move $s1, $v0


li $v0,4
la $a0, y1
syscall
```

```
li $v0,5    #user to input the y1 number and save it in $s2
syscall
move $s2, $v0


li $v0,4
la $a0, y2
syscall

li $v0,5      #user to input the y2 number and save it in $s3
syscall
move $s3, $v0

li $v0,4
la $a0, newdistance
syscall


sub $t0,$s0,$s3    #subtraction x1 and y1
sub    $t1,$s1,$s3    # subtracting x2 and y2
mul $t0,$t0,$t0    #squaring the result of subtraction of x1 and y1
mul $t1,$t1,$t1 #squaring the result of subtraction of x2 and y2
add $t3, $t0, $t1      # adding both results
li $s0,1 # i =1
move $t4,$t3 # saving the above value into $t4
while:
bgt $t4,$s0, change    #while $t4>$s0 change the value of $t4
ble $t4,$s0,exit              #if its less then print the value saved in $t4
change:
add $t4, $t4, $s0 #n+i
li $s1, 2
div $t4, $t4, $s1
div $s0, $t3,$t4
j while
exit:
move $a0,$t4
li $v0,1    # to print the answer in integer
syscall
j main
```

FLOWCHART FOR HAMMING DISTANCE



FLOWCHART FOR EUCLIDEAN DISTANCE



3. COMMENTS FOR PROJECT (5 POINTS)

The project was fairly easy. Calculation and the Euclidean distance wasn't hard. However I had little trouble with the hamming distance because of the loops and while statements. But we had seen it in the course in exams and homework's so after reviewing I could do it. Overall I would say the project was interesting as I learned new stuff and much experience working on mips.