**Assignment 3:**

* **Set1-20180080**

1. **Problem 1**
2. .text
3. li $s3 ,1
4. li $v0 , 5
5. syscall
6. addi $s0 , $v0 ,0
7. li $v0 , 5
8. syscall
9. addi $s1, $v0 ,0
11. slt $t0 , $s1 , $s0
12. bne $t0,1,else1
14. add $s0, $s0 ,$s1
15. j exit1
16. else1: bne $s0, $s1,else2
18. mul $s0, $s0 ,$s1
19. j exit1
20. else2: sub $s0, $s0 ,$s1
22. exit1:
24. li $v0 , 10
25. syscall

**2.Problem 2**

.data #here we declare variables

char\_arr: .space 20

upper: .asciiz "\n num of upper is "

lower: .asciiz "\n num of lower is "

.text

.globl main

main:

# t3=char t2 =location of char t0 =check less than t1=i

la $t2 , char\_arr

li $t1 , 0

li $t4,0

li $t5,0

for:

slti $t0 ,$t1 , 5

beq $t0 , 0 , exitfor

# reads a character

li $v0, 12

syscall

addi $t3 , $v0 ,0

sw $t3 , ($t2)

slti $t0 , $t3 , 65

beq $t0 , 1 , exit1

slti $t0 , $t3, 90

beq $t0 , 0 , exit1

addi $t4 , $t4 ,1

exit1:

slti $t0 , $t3 , 97

beq $t0 , 1 , exit2

slti $t0 , $t3, 122

beq $t0 , 0 , exit2

addi $t5 , $t5 ,1

exit2:

addi $t2 , $t2 , 4

addi $t1 , $t1 , 1

j for

exitfor:

li $v0, 4

la $a0, upper

syscall

li $v0 ,1

addi $a0 , $t4 , 0

syscall

li $v0, 4

la $a0, lower

syscall

li $v0,1

addi $a0 , $t5,0

syscall

li $v0, 10

syscall

**3. Problem 3**

.data

arr: .space 16

.text

main:

la $t0 , arr

li $t5 , 0

sw $t5 , ($t0)

sw $t5 , 4($t0)

sw $t5 , 8($t0)

sw $t5 , 12($t0)

li $t1 , 0

for1:slti $t3 , $t1 , 4

beq $t3 , 0 , exit1

li $t2 , 0

for2:slti $t6, $t2 , 4

beq $t6, 0 , exit2

sub $t7 , $t1 , $t2

addi $t2 , $t2 , 1

j for2

exit2:addi $t1 , $t1, 1

sw $t7 , ($t0)

addi $t0 , $t0, 8

j for1

exit1:

.end main

li $v0 , 10

syscall

1. **Problem 4**

.text

.globl main

main:

li $a3 , 10

jal sumodd

li $v0 ,1

addi $a0 , $s1 , 0

syscall

li $v0,10

syscall

sumodd:

add $t0 , $zero ,$zero

#li $v1 , 0

add $t1 , $zero ,$zero

addi $t2 , $zero ,1

div $a3 , $a3 , 2

while:

slt $t3 , $t0 , $a3

beq $t3 , 0 , exitwhile

add $t1 , $t1 , $t2

add $s1 , $t1, $zero

addi $t2 , $t2 , 2

addi $t0 , $t0 ,1

j while

exitwhile:

jr $ra

.end main

li $v0 , 10

syscall

1. **Problem 5**

.data

newline: .asciiz "\n"

space: .asciiz " "

.text

.globl main

main:

#0x3fc00000

#0x4050000

#li $v0, 5 # service 5 (read integer)

#add $s4, $zero, $v0

#syscall

#li $v0, 5 # service 5 (read integer)

#add $s5, $zero, $v0

#syscall

#li $s4 ,2192048128 #1 00000101 01010000000000000000000

#li $s5 ,53477376 #0 00000110 01100000000000000000000

#0 00000000 11111111111111111111111

li $s4 ,0x81B00000 #1 00000011 01100000000000000000000

li $s5 ,0x2B80000 #0 00000101 01110000000000000000000

#sign

srl $t1, $s4 , 31 #1

srl $t2, $s5 , 31 #0

xor $s0 , $t1 , $t2 #1

sll $s0 , $s0 , 31 #1 00000000 00000000000000000000000

#-----------------------------exponet----------------------------

srl $t4 , $s4 , 23 # 00000011

srl $t5 , $s5 , 23 # 00000101

andi $t4 , $t4 , 255 #00000011 and 011111111 == 101 == 5

andi $t5 , $t5 , 255 #00000101 and 011111111 == 101 == 5

addi $t4 , $t4 , -127

addi $t5 , $t5 , -127

add $s1 , $t4 , $t5

add $s1 , $s1 , 127 #110001001

sll $s1 , $s1 , 23 #110001001 00000000000000000000000

#-----------------------------------mantessa-------------------------------

andi $t6 , $s4 ,8388607 #0 00000000 11111111111111111111111

andi $t7 , $s5 , 8388607

ori $s6 , $t6 ,0x00800000 #1 00000000000000000000000 add1 in last bit

ori $s7 , $t7 ,0x00800000 #1 00000000000000000000000 add1 in last bit

mult $s6 , $s7

mfhi $t2 #11111010000000000000000

#-------------------normalization------------------

sll $t2 , $t2 ,16

norm:

li $s6, 31 #counter

li $t8 , 1

srlv $t9 , $t2 ,$s6

li $v0 , 1

add $a0 , $t9 , $0

syscall

ori $t9 , $t9 ,1

li $v0 , 1

add $a0 , $t9 , $0

syscall

subi $s6 , $s6 , 1

li $v0 , 1

add $a0 , $s6 , $0

syscall

addi $t8 , $t8 ,1

beqz $t9 , norm

li $v0 , 1

add $a0 , $t9 , $0

syscall

beqz $s6 , res

li $v0 , 1

add $a0 , $s6 , $0

syscall

res:

sllv $t2 , $t2 , $t8

srl $t2 , $t2 , 9

or $s3 , $s0 , $s1

or $s3 , $s3 ,$t2

li $v0 , 10

syscall

**--------------------------------------------------------------**

.data

a: .float 1.123

b: .float 1.5

.text

main :

lwc1 $f2 , a

lwc1 $f4 , b

add.s $f12,$f2 , $f4

#li $v0 , 3

#syscall

li $t1 , 5

div $t0 ,$t1 ,2

li $v0 , 1

addi $a0 , $t0 , 0

syscall

li $v0 , 10

syscall

* **Set2-20180104**
* **Set3-20180080 & 20180104**

1. **Problem 1**
2. **Problem 2**

.data

str: .space 20

str2: .space 20

.text

.globl main

main:

add $t0, $zero, $zero #t0 = i counter for the loops

add $t1, $zero, $zero #t1 = j counter for length of the string

li $v0, 8 #gets(str)

la $a0, str

la $a1, 20

syscall

li $t5 ,0

length:

lb $s0, str($t0) #load each character to s0

beq $s0, '\n', NEXTCHECK

add $t0, $t0, 1 #i++ to scan all the characters of the string

add $t1, $t1, 1 #j++ for the total length of the string

j length

NEXTCHECK :

add $t0, $zero, $zero #clean the t0 register from the length loop

addi $t4 , $t1 , -1

for:

slt $t3, $t0, $t1 #for(i=0; i<length; i++)

beq $t3,0, EXIT

lb $s1, str($t0) #str[i]

sb $s1, str2($t4) #str[length-i-1]

addi $t4, $t4, -1

addi $t0, $t0, 1 #i++

j for

EXIT:

li $v0, 4 #function to print string

la $a0, str2 #load address of message into register $a0

syscall

li $v0 , 10

syscall

1. **Problem 3 by 20180080**

.data

arr: .space 16

.text

main:

la $t0 , arr

li $t5 , 1

sw $t5 , ($t0)

sw $t5 , 4($t0)

sw $t5 , 8($t0)

sw $t5 , 12($t0)

li $t1 , 0

for1:slti $t3 , $t1 , 4

beq $t3 , 0 , exit1

addi $t2 ,$t1, 0

for2:slti $t6, $t2 , 0

beq $t6, 1 , exit2

lw $t7 , ($t0)

mul $t7 , $t7 , $t2

subi $t2 , $t2 , 1

j for2

exit2:

addi $t1 , $t1, 1

sw $t7 , ($t0)

addi $t0 , $t0, 4

j for1

exit1:

.end main

1. **Problem 4**
2. **Problem 5 by 20180080**

.data

str: .space 20

a: .asciiz "it is not palindrom!"

b: .asciiz "it is palindrom!"

.text

.globl main

main:

add $t0, $zero, $zero #t0 = i counter for the loops

add $t1, $zero, $zero #t1 = j counter for length of the string

li $v0, 8 #gets(str)

la $a0, str

la $a1, 20

syscall

li $t5 ,0

length:

lb $s0, str($t0) #load each character to s0

beq $s0, '\n', NEXTCHECK

add $t0, $t0, 1 #i++ to scan all the characters of the string

add $t1, $t1, 1 #j++ for the total length of the string

j length

NEXTCHECK :

add $t0, $zero, $zero #clean the t0 register from the length loop

div $s2 , $t1 , 2

addi $t4 , $t1 , -1

div $t1 , $t1 ,2

for:

slt $t3, $t0, $t1 #for(i=0; i<length; i++)

beq $t3,0, EXIT

lb $s0, str($t0) #str[i]

lb $s1, str($t4) #str[length-i-1]

addi $t4, $t4, -1

addi $t0, $t0, 1 #i++

bne $s0, $s1, A #if (str[i] == str[length-i-1])

addi $t5 , $t5 ,1

A:

j for

EXIT:

beq $s2 , $t5 , B

li $v0, 4 #rintf("It is not palindromic");

la $a0, a

syscall

j EXITi

B:li $v0, 4 #printf("It is palindromic");

la $a0, b

syscall

EXITi:

li $v0 , 10

syscall