**BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH (IT3280) GIỮA KỲ**

***Họ và tên: Phạm Đức Long***

***MSSV: 20225737***

Assignment 1:

* Code:

.eqv SEVENSEG\_RIGHT 0xFFFF0010

.eqv SEVENSEG\_LEFT 0xFFFF0011

.textmain:

li $a0, 79 # hien thi so 3

jal SHOW\_7SEG\_LEFT

li $a0, 7 # hien thi so 7

jal SHOW\_7SEG\_RIGHT

exit:

li $v0, 10 # ket thuc chuong trinh

syscall

endmain:

SHOW\_7SEG\_LEFT:

li $t0, SEVENSEG\_LEFT

sb $a0, 0($t0)

jr $ra

SHOW\_7SEG\_RIGHT:

li $t0, SEVENSEG\_RIGHT

sb $a0, 0($t0)

jr $ra

* Kết quả thu được:

A screenshot of a computer

Description automatically generated

* Nhận xét: Kết quả thu được là 37, hoàn toàn trùng khớp với hai số cuối của Mã số sinh viên 20225737. Do đó kết quả thu được hoàn toàn chính xác

Assignment 2:

* Code:

.eqv SEVENSEG\_LEFT 0xFFFF0010

.eqv SEVENSEG\_RIGHT 0xFFFF0011

.data

Message: .asciiz "Nhap so nguyen: "

.text

main:

# nhap so nguyen tu ban phim:

li $v0, 4

la $a0, Message

syscall

li $v0, 5

syscall

# kiem tra so ben trai va so ben phai (hang chuc va hang don vi):

div $v0, $v0, 10

mfhi $a0

jal CHECK

jal SHOW\_7SEG\_LEFT

div $v0, $v0, 10

mfhi $a0

jal CHECK

jal SHOW\_7SEG\_RIGHT

exit:

# ket thuc chuong trinh

li $v0, 10

syscall

endmain:

CHECK:

# kiem tra xem gia tri cua so la bao nhieu:

beq $a0, 0, v0

beq $a0, 1, v1

beq $a0, 2, v2

beq $a0, 3, v3

beq $a0, 4, v4

beq $a0, 5, v5

beq $a0, 6, v6

beq $a0, 7, v7

beq $a0, 8, v8

beq $a0, 9, v9

# neu la cac so (tu 0 den 9) thi sau do hien thi tren led 7 thanh:

v0:

addi $a0, $zero, 63

jr $ra

v1:

addi $a0, $zero, 6

jr $ra

v2:

addi $a0, $zero, 91

jr $ra

v3:

addi $a0, $zero, 79

jr $ra

v4:

addi $a0, $zero 102

jr $ra

v5:

addi $a0, $zero, 109

jr $ra

v6:

addi $a0, $zero, 125

jr $ra

v7:

addi $a0, $zero, 7

jr $ra

v8:

addi $a0, $zero, 127

jr $ra

v9:

addi $a0, $zero, 111

jr $ra

SHOW\_7SEG\_LEFT:

li $t0, SEVENSEG\_LEFT

sb $a0, 0($t0)

jr $ra

SHOW\_7SEG\_RIGHT:

li $t0, SEVENSEG\_RIGHT

sb $a0, 0($t0)

jr $ra

* Kết quả thu được:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Nhận xét: Các kết quả thu được từ Led 7 thanh đều trùng khớp với số nguyên nhập vào từ bàn phím.

Assignment 3:

* Code:

.eqv SEVENSEG\_LEFT 0xFFFF0010

.eqv SEVENSEG\_RIGHT 0xFFFF0011

.data

Message: .asciiz "Nhap ky tu: "

.text

main:

# nhap vao mot ky tu tu ban phim:

li $v0, 4

la $a0, Message

syscall

li $v0, 12

syscall

div $v0, $v0, 10

mfhi $a0

jal CHECK

jal SHOW\_7SEG\_LEFT

div $v0, $v0, 10

mfhi $a0

jal CHECK

jal SHOW\_7SEG\_RIGHT

exit:

# ket thuc chuong trinh

li $v0, 10

syscall

endmain:

CHECK:

# kiem tra xem gia tri cua so la bao nhieu:

beq $a0, 0, v0

beq $a0, 1, v1

beq $a0, 2, v2

beq $a0, 3, v3

beq $a0, 4, v4

beq $a0, 5, v5

beq $a0, 6, v6

beq $a0, 7, v7

beq $a0, 8, v8

beq $a0, 9, v9

# neu la cac so (tu 0 den 9) thi sau do hien thi tren led 7 thanh:

v0:

addi $a0, $zero, 63

jr $ra

v1:

addi $a0, $zero, 6

jr $ra

v2:

addi $a0, $zero, 91

jr $ra

v3:

addi $a0, $zero, 79

jr $ra

v4:

addi $a0, $zero 102

jr $ra

v5:

addi $a0, $zero, 109

jr $ra

v6:

addi $a0, $zero, 125

jr $ra

v7:

addi $a0, $zero, 7

jr $ra

v8:

addi $a0, $zero, 127

jr $ra

v9:

addi $a0, $zero, 111

jr $ra

SHOW\_7SEG\_LEFT:

li $t0, SEVENSEG\_LEFT

sb $a0, 0($t0)

jr $ra

SHOW\_7SEG\_RIGHT:

li $t0, SEVENSEG\_RIGHT

sb $a0, 0($t0)

jr $ra

* Kết quả thu được:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

* Giải thích: Ký tự ‘d’ có giá trị ascii là 100, nên LED hiển thị 00, còn ‘x’ có giá trị 120 nên LED hiện thị 20.
* Nhận xét: Kết quả thu được hoàn toàn đúng với giá trị trong bảng mã ascii.

Assignment 4:

* Code:

.eqv MONITOR\_SCREEN 0x10010000

.eqv RED 0x00FF0000

.eqv YELLOW 0x00FFFF00

.text

li $k0, MONITOR\_SCREEN

addi $k1, $k0, 256 # co 64 o => phai co 4 \* 64 = 256 lan nhay

addi $a0, $zero, 0

LOOP:

beq $k0, $k1, END

beq $a0, 4, REVERSE # $a0 = 4 thi xuong hang duoi

li $t0, YELLOW

sw $t0, 0($k0)

li $t0, RED

sw $t0, 4($k0) # cach mot o mau vang la mot o mau do

addi $a0, $a0, 1 # $a0++

addi $k0, $k0, 8 # $k0 +=8 de to hai o tiep theo

j LOOP

REVERSE:

# xuong hang duoi va dao nguoc mau so voi cac hang ben tren (vang -> do, do -> vang)

beq $k0, $k1, END

beqz $a0, LOOP # $a0 = 0 thi xuong hang duoi

li $t0, RED

sw $t0, 0($k0)

li $t0, YELLOW

sw $t0, 4($k0)

addi $a0, $a0, -1 # $a0--

addi $k0, $k0, 8 # $k0 +=8 de to hai o tiep theo

j REVERSE

END:

# ket thuc chuong trinh

li $v0, 10

syscall

* Kết quả thu được:

A screenshot of a computer

Description automatically generated

Assignment 5:

* Code:

.eqv MONITOR\_SCREEN 0x10010000

.eqv RED 0x00FF0000

.eqv GREEN 0x0000FF00

.data

Message\_x1: .asciiz "Nhap vao x1: "

Message\_y1: .asciiz "Nhap vao y1: "

Message\_x2: .asciiz "Nhap vao x2: "

Message\_y2: .asciiz "Nhap vao y2: "

.text

# nhap cac gia tri x1, y1, x2, y2

li $v0, 4

la $a0, Message\_x1

syscall

li $v0, 5

syscall

move $t1, $v0 # gan $t1 = x1

li $v0, 4

la $a0, Message\_y1

syscall

li $v0, 5

syscall

move $s1, $v0 # gan $s1 = y1

li $v0, 4

la $a0, Message\_x2

syscall

li $v0, 5

syscall

move $t2, $v0 # gan $t2 = x2

li $v0, 4

la $a0, Message\_y2

syscall

li $v0, 5

syscall

move $s2, $v0 # gan $s2 = y2

MONITOR:

# chinh sua cac gia tri

li $k0, MONITOR\_SCREEN

li $t0, RED

addi $v0, $zero, 0

addi $t3, $t1, -1 # x1--

addi $s3, $s1, -1 # y1--

addi $t4, $t2, 1 # x2++

addi $s4, $s2, 1 # y2++

# to mau hinh chu nhat

BORDER:

mul $k1, $s4, 8

add $k1, $k1, $t4

mul $k1, $k1, 4

add $k1, $k1, $k0

# goc phai ben duoi

LOOP:

mul $a0, $s3, 8

add $a0, $a0, $t3

mul $a0, $a0, 4

add $a0, $a0, $k0

# goc trai ben tren

beq $a0, $k1, RESET\_BORDER # $a0 = $k1 thi ve duong vien

beq $t3, $t4, RESET # x1 = x2 thi reset de ve tiep hinh chu nhat

sw $t0, 0($a0) # to mau do

addi $t3, $t3, 1

j LOOP

RESET:

addi $t3, $t1, -1 # reset lai x1 ve lai tu dau

add $t3, $t3, $v1

addi $s3, $s3, 1 # tang y1 len 1 de xuong dong

j LOOP

# to mau vien

RESET\_BORDER:

bnez $v1, EXIT

addi $v1, $v1, 1 # tang $v1 len 1 den ket thuc chuong trinh voi lenh bnez o tren

li $t0, GREEN

addi $t3, $t1, 0

addi $s3, $s1, 0

addi $t4, $t2, 0

addi $s4, $s2, 0

j BORDER

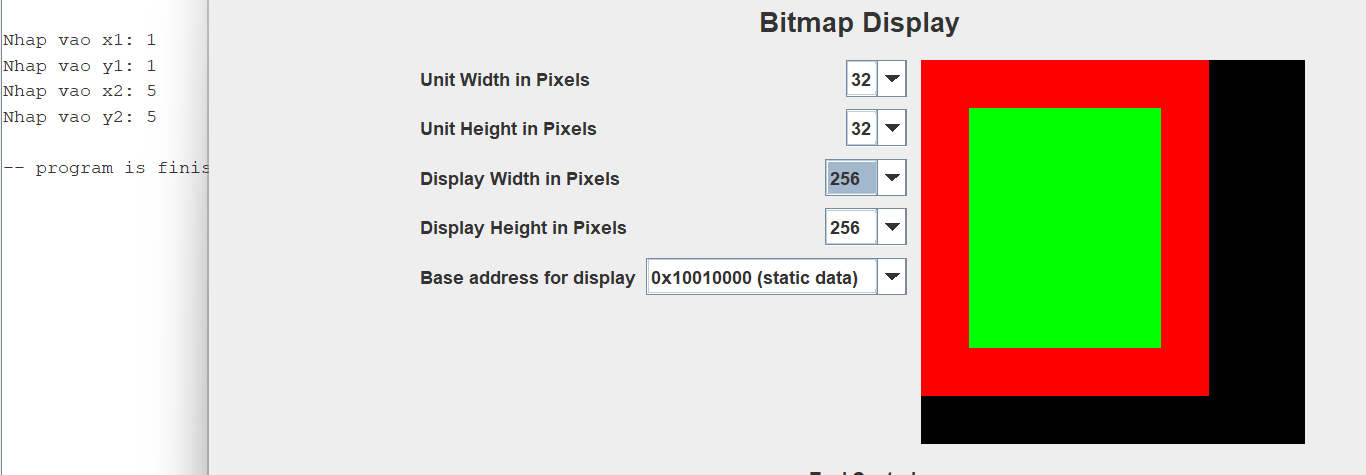
EXIT:

# ket thuc chuong trinh

li $v0, 10

syscall

* Kết quả thu được:



* Nhận xét: Kết quả thu được là Bitmap hoàn toàn đúng với dữ kiện nhập vào từ bàn phím