

Bài thực hành số 10

Họ tên: Nguyễn Văn Hiếu -20225717

Assignment 1:

```
.eqv SEVENSEG_LEFT  0xFFFF0011 # Địa chỉ của đèn led 7 đoạn trái.  
                        #   Bit 0 = đoạn a;  
                        #   Bit 1 = đoạn b; ...  
                        #   Bit 7 = dấu .  
  
.eqv SEVENSEG_RIGHT 0xFFFF0010 # Địa chỉ của đèn led 7 đoạn phải
```

```
.text
```

```
main:
```

```
    li $a0, 0x06    # set value for segments  
    jal SHOW_7SEG_LEFT    # show  
    nop  
  
    li $a0, 0x07    # set value for segments  
    jal SHOW_7SEG_RIGHT    # show  
    nop
```

```
exit: li $v0, 10
```

```
    syscall
```

```
endmain:
```

```
#-----
```

```
# Function SHOW_7SEG_LEFT : turn on/off the 7seg
```

```
# param[in] $a0  value to shown
```

```
# remark  $t0 changed
```

```
#-----
```

```
SHOW_7SEG_LEFT: li  $t0, SEVENSEG_LEFT # assign port's address
```

```
    sb  $a0, 0($t0)    # assign new value
```

```
    nop
```

```
    jr  $ra
```

```
    nop
```

```
#-----
```

```
# Function SHOW_7SEG_RIGHT : turn on/off the 7seg
```

```
# param[in] $a0  value to shown
```

```
# remark  $t0 changed
```

```
#-----
```

```
SHOW_7SEG_RIGHT: li  $t0, SEVENSEG_RIGHT # assign port's address
```

```
    sb  $a0, 0($t0)    # assign new value
```

```
    nop
```

```
    jr  $ra
```

```
    nop
```

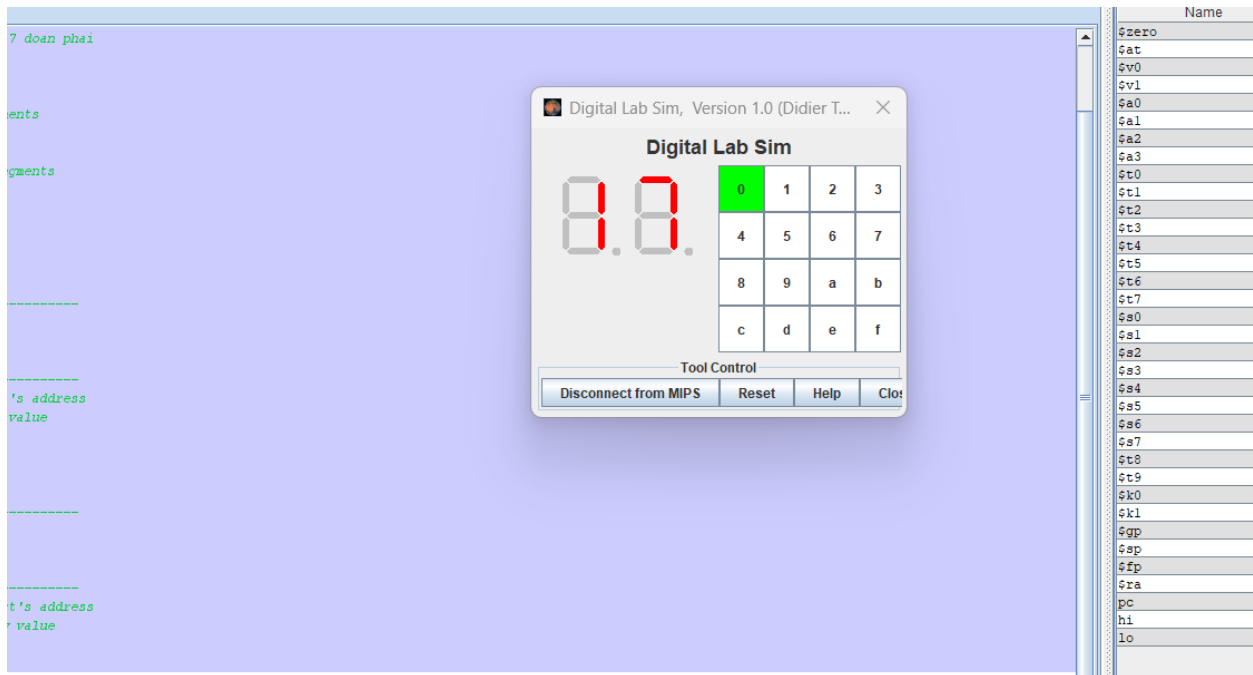
Đoạn code trên dùng để hiển thị ra 2 chữ cuối MSSV của mình trên led 7 thanh.

Mssv 20225717 ➔ Hiển thị ra số số 17 trên led 7 thanh.

Để hiển thị số 1 ứng với thanh b và c sáng → \$a0=0x06

Để hiển thị số 7 ứng với thanh a,b và c sáng → \$a0=0x07

Kết quả:



Assginment 2:

.data

message: .asciiz "Nhap 1 so bat ki: "

.eqv SEVENSEG_LEFT 0xFFFF0011 # Dia chi cua den led 7 doan trai.

Bit 0 = doan a;

Bit 1 = doan b; ...

Bit 7 = dau .

.eqv SEVENSEG_RIGHT 0xFFFF0010 # Dia chi cua den led 7 doan phai

.text

main:

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

li $v0, 5      #Nhap 1 gia tri tu ban phim
syscall

move $s0,$v0   #Luu gia tri vao thanh ghi s0

li $t4,100

div $s0,$t4     #Chia lay du so nhap vao cho 100 de lay 2 so cuoi
mfhi $s1        #Luu gia tri tim duoc vao thanh ghi s1

li $t5,10

div $s1,$t5     #Lay chu so cuoi cung
mfhi $s2        #Luu gia tri vao thanh ghi s2
mflo $s3

beq $s3,0,case_0
beq $s3,1,case_1
beq $s3,2,case_2
beq $s3,3,case_3
beq $s3,4,case_4
beq $s3,5,case_5
beq $s3,6,case_6
beq $s3,7,case_7
beq $s3,8,case_8
beq $s3,9,case_9
```

```
case_0: li $a2,0x3F
j continue
case_1: li $a2,0x03
j continue
case_2: li $a2,0x5B
j continue
case_3: li $a2,0x4F
j continue
case_4: li $a2, 0x66
j continue
case_5: li $a2,0x6D
j continue
case_6: li $s2,0x7D
j continue
case_7: li $a2,0x07
j continue
case_8: li $a2, 0x7F
j continue
case_9: li $a2, 0x6F
j continue

        continue:      # set value for segments
jal  SHOW_7SEG_LEFT    # show
```

```
    nop
    beq $s2,0,case_00
    beq $s2,1,case_11
    beq $s2,2,case_22
    beq $s2,3,case_33
    beq $s2,4,case_44
    beq $s2,5,case_55
    beq $s2,6,case_66
    beq $s2,7,case_77
    beq $s2,8,case_88
    beq $s2,9,case_99
case_00: li $a3,0x3F
j continue1
case_11: li $a3,0x03
j continue1
case_22: li $a3,0x5B
j continue1
case_33: li $a3,0x4F
j continue1
case_44: li $a3, 0x66
j continue1
case_55: li $a3,0x6D
j continue1
```

```
case_66: li $s3,0x7D
```

```
j continue1
```

```
case_77: li $a3,0x07
```

```
j continue1
```

```
case_88: li $a3, 0x7F
```

```
j continue1
```

```
case_99: li $a3, 0x6F
```

```
j continue1
```

```
continue1:
```

```
    jal SHOW_7SEG_RIGHT    # show
```

```
    nop
```

```
exit:
```

```
    syscall
```

```
    li $v0, 10
```

```
    syscall
```

```
endmain:
```

```
#-----
```

```
# Function SHOW_7SEG_LEFT : turn on/off the 7seg
```

```
# param[in] $a0  value to shown
```

```
# remark  $t0 changed
```

```
#-----
```

```
SHOW_7SEG_LEFT: li $t0, SEVENSEG_LEFT # assign port's address
```

```
sb $a2, 0($t0)    # assign new value
```

```
nop
```

```
jr $ra
```

```
nop
```

```
#-----
```

```
# Function SHOW_7SEG_RIGHT : turn on/off the 7seg
```

```
# param[in] $a0  value to shown
```

```
# remark  $t0 changed
```

```
#-----
```

```
SHOW_7SEG_RIGHT: li $t0, SEVENSEG_RIGHT # assign port's address
```

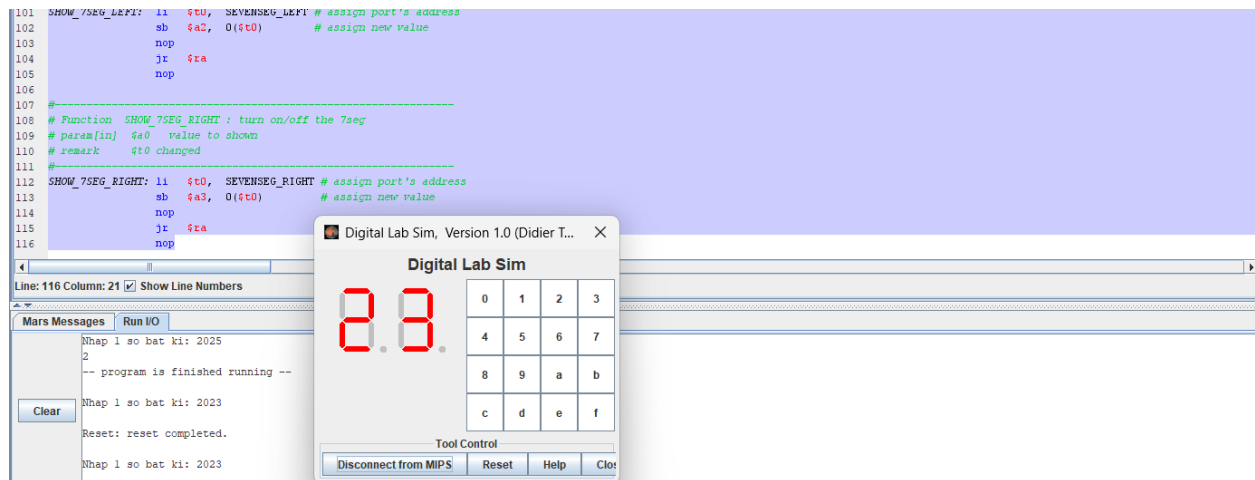
```
sb $a3, 0($t0)    # assign new value
```

```
nop
```

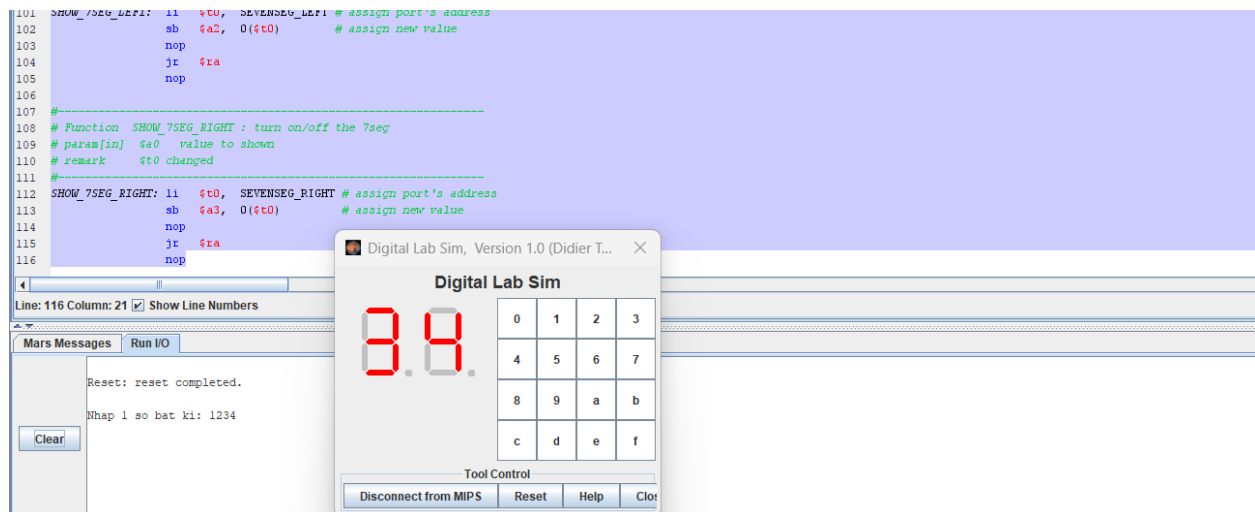
```
jr $ra
```

```
nop
```

Giả sử khi nhập vào số 2023. Kết quả thu được:



Khi nhập vào số 1234. Kết quả thu được:



Assginment 3:

.data

message: .ascii "Nhập 1 ki tu bất kỳ: "

.eqv SEVENSEG_LEFT 0xFFFF0011 # Địa chỉ của đèn led 7 đoạn trái.

Bit 0 = đoạn a;

Bit 1 = đoạn b; ...

Bit 7 = dấu .

.eqv SEVENSEG_RIGHT 0xFFFF0010 # Dia chi cua den led 7 doan phai

.text

main:

li \$v0,4

la \$a0,message

syscall

li \$v0, 12 #Nhap 1 gia tri tu ban phim

syscall

move \$s0,\$v0 #Luu gia tri vao thanh ghi s0

li \$t4,100

div \$s0,\$t4 #Chia lay du so nhap vao cho 100 de lay 2 so cuoi

mfhi \$s1 #Luu gia tri tim duoc vao thanh ghi s1

li \$t5,10

div \$s1,\$t5 #Lay chu so cuoi cung

mfhi \$s2 #Luu gia tri vao thanh ghi s2

mflo \$s3

beq \$s3,0,case_0

beq \$s3,1,case_1

beq \$s3,2,case_2

beq \$s3,3,case_3

beq \$s3,4,case_4

beq \$s3,5,case_5

```
        beq $s3,6,case_6
        beq $s3,7,case_7
        beq $s3,8,case_8
        beq $s3,9,case_9
case_0: li $a2,0x3F
j continue
case_1: li $a2,0x03
j continue
case_2: li $a2,0x5B
j continue
case_3: li $a2,0x4F
j continue
case_4: li $a2, 0x66
j continue
case_5: li $a2,0x6D
j continue
case_6: li $a2,0x7D
j continue
case_7: li $a2,0x07
j continue
case_8: li $a2, 0x7F
j continue
case_9: li $a2, 0x6F
```

```
j continue

    continue:      # set value for segments
jal  SHOW_7SEG_LEFT    # show

nop

    beq $s2,0,case_00
    beq $s2,1,case_11
    beq $s2,2,case_22
    beq $s2,3,case_33
    beq $s2,4,case_44
    beq $s2,5,case_55
    beq $s2,6,case_66
    beq $s2,7,case_77
    beq $s2,8,case_88
    beq $s2,9,case_99

case_00: li $a3,0x3F
j continue1

case_11: li $a3,0x03
j continue1

case_22: li $a3,0x5B
j continue1

case_33: li $a3,0x4F
j continue1

case_44: li $a3, 0x66
```

```
j continue1
case_55: li $a3,0x6D
j continue1
case_66: li $a3,0x7D
j continue1
case_77: li $a3,0x07
j continue1
case_88: li $a3, 0x7F
j continue1
case_99: li $a3, 0x6F
j continue1
continue1:
    jal SHOW_7SEG_RIGHT    # show
    nop
```

exit:

```
    syscall
    li $v0, 10
    syscall
```

endmain:

```
#-----
# Function SHOW_7SEG_LEFT : turn on/off the 7seg
# param[in] $a0  value to shown
```

```
# remark $t0 changed
```

```
#-----
```

```
SHOW_7SEG_LEFT: li $t0, SEVENSEG_LEFT # assign port's address
```

```
sb $a2, 0($t0) # assign new value
```

```
nop
```

```
jr $ra
```

```
nop
```

```
#-----
```

```
# Function SHOW_7SEG_RIGHT : turn on/off the 7seg
```

```
# param[in] $a0 value to shown
```

```
# remark $t0 changed
```

```
#-----
```

```
SHOW_7SEG_RIGHT: li $t0, SEVENSEG_RIGHT # assign port's address
```

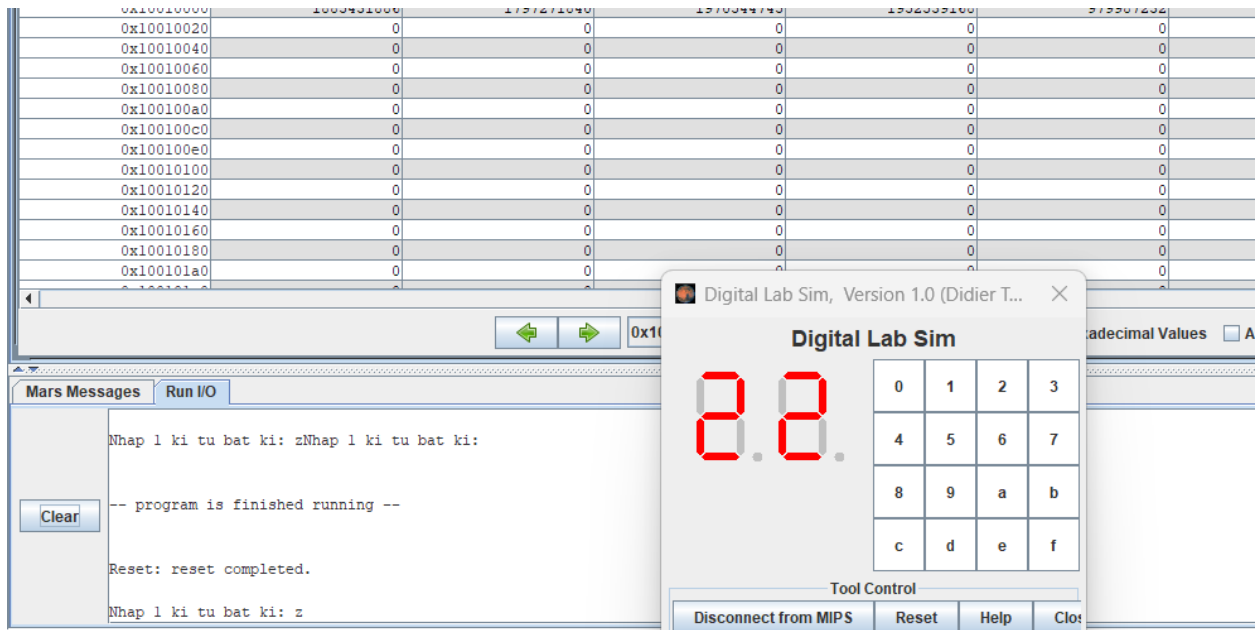
```
sb $a3, 0($t0) # assign new value
```

```
nop
```

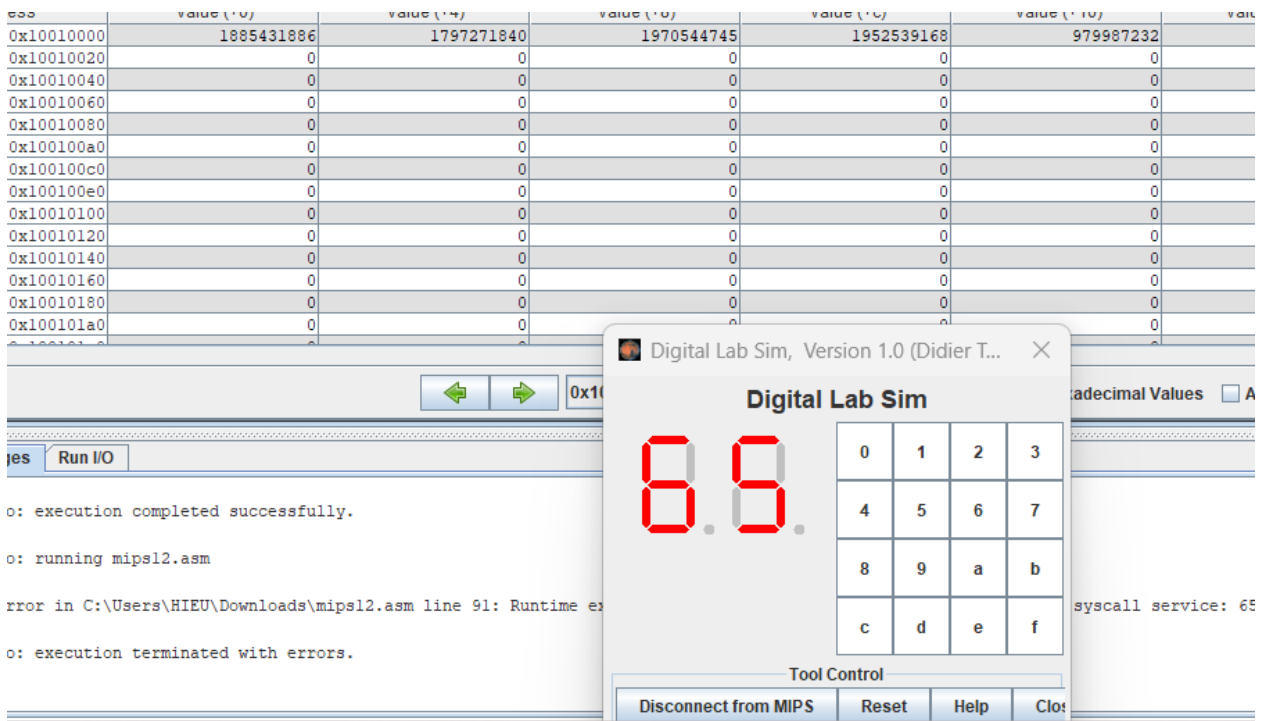
```
jr $ra
```

```
nop
```

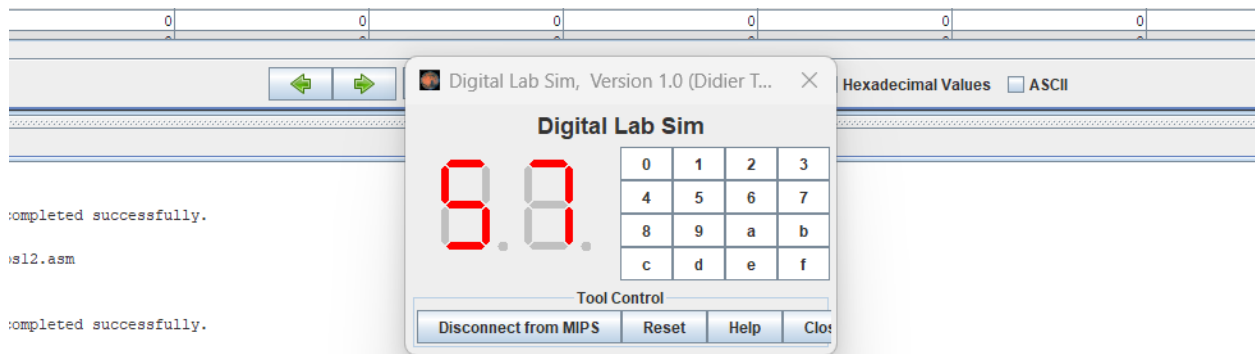
Khi nhập vào kí tự z có giá trị 122 kết quả là:



Khi nhập giá trị A có giá trị 65. Kết quả là:



Khi nhập số 9 có giá trị là 57. Kết quả là:



Assignment 4:

```
.eqv MONITOR_SCREEN 0x10010000
```

```
.eqv RED 0x00FF0000
```

```
.eqv YELLOW 0x00FFFF00
```

```
.text
```

```
li $k0, MONITOR_SCREEN
```

```
li $s0, 2
```

```
li $t0, -1 # Khoi tao j
```

```
For1: addi $t0, $t0, 1
```

```
beq $t0, 8, Exit
```

```
li $t1, -1 # Khoi tao i
```

```
For2: addi $t1, $t1, 1
```

```
beq $t1, 8, EndFor2
```

```
div $t0, $s0
```

```
mfhi $t2
```

```
div $t1, $s0
```

```
mfhi $t3
```



```
bne $t2, 0, Next  
bne $t3, 0, Paint2  
j Paint1
```

Next:

```
beq $t3, 0, Paint2
```

Paint1:

```
sll $s1, $t0, 3  
add $s1, $s1, $t1  
sll $s1, $s1, 2  
add $s2, $s1, $k0  
li $t4, RED  
sw $t4, 0($s2)
```

```
j For2
```

Paint2:

```
sll $s1, $t0, 3  
add $s1, $s1, $t1  
sll $s1, $s1, 2  
add $s2, $s1, $k0  
li $t4, YELLOW  
sw $t4, 0($s2)
```

```
j For2
```

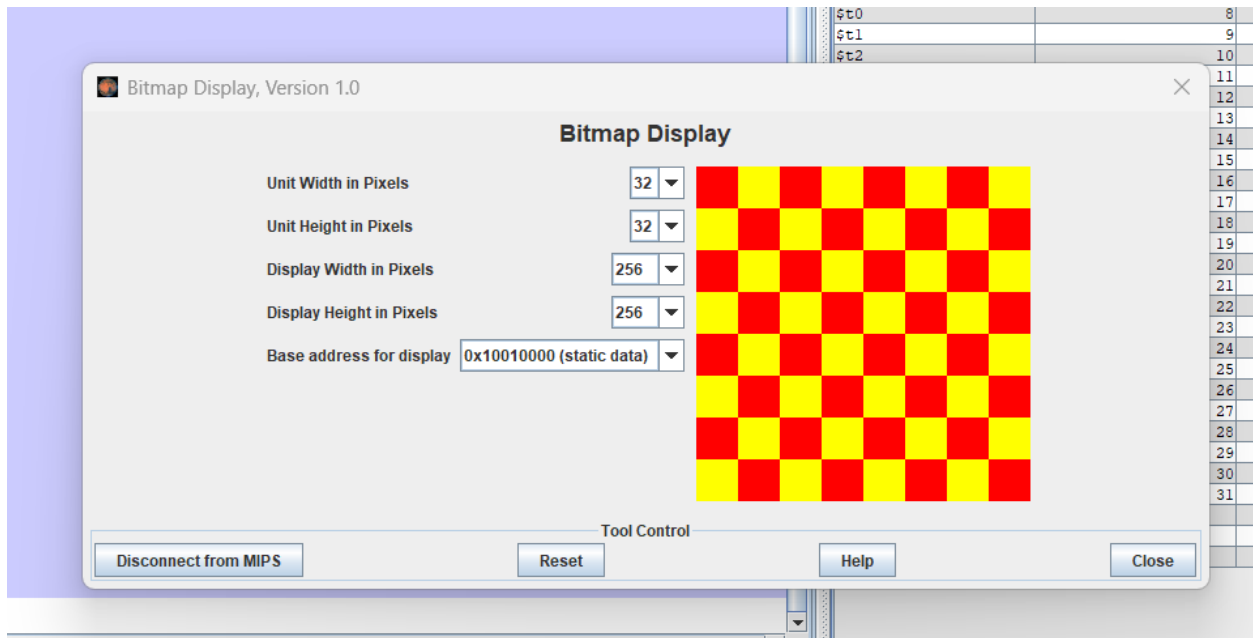
EndFor2:

```
j For1
```

Exit: li \$v0, 10

Syscall

Kết quả thu được:



Assignment 5:

```
.eqv MONITOR_SCREEN 0x10010000
```

```
.eqv RED 0x00FF0000
```

```
.eqv GREEN 0x0000FF00
```

```
.data
```

```
x1: .asciiz "Nhap x1: "
```

```
y1: .asciiz "Nhap y1: "
```

```
x2: .asciiz "Nhap x2: "
```

```
y2: .asciiz "Nhap y2: "
```

```
error1: .asciiz "Error: x2 phai khac x1. Moi nhap lai!\n"
```

```
error2: .asciiz "Error: y2 phai khac y1. Moi nhap lai!\n"
```

.text

li \$k0, MONITOR_SCREEN

li \$v0, 4

la \$a0, x1

syscall

li \$v0, 5

syscall

move \$s0, \$v0

li \$v0, 4

la \$a0, y1

syscall

li \$v0, 5

syscall

move \$s1, \$v0

NhapX2: li \$v0, 4

la \$a0, x2

syscall

li \$v0, 5

syscall

move \$s2, \$v0

beq \$s2, \$s0, Error1

NhapY2: li \$v0, 4

la \$a0, y2

syscall

li \$v0, 5

syscall

move \$s3, \$v0

beq \$s3, \$s1, Error2

j Tsugi

Error1: li \$v0, 4

la \$a0, error1

syscall

j NhapX2

Error2: li \$v0, 4

la \$a0, error2

syscall

j NhapY2

Tsugi:

slt \$t0, \$s0, \$s2

slt \$t1, \$s1, \$s3

beq \$t0, 0, Case3

beq \$t1, 0, Case2

Case1:

add \$v0, \$s1, \$zero

For1:

bgt \$v0, \$s3, Exit

add \$v1, \$s0, \$zero

For2:

bgt \$v1, \$s2, EndFor2

beq \$v0, \$s1, InVien1

beq \$v0, \$s3, InVien1

beq \$v1, \$s0, InVien1

beq \$v1, \$s2, InVien1

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, GREEN

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For2

InVien1:

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

```
li $a1, RED
add $a2, $k0, $t8
sw $a1, 0($a2)
add $v1, $v1, 1
j For2
EndFor2:
add $v0, $v0, 1
j For1
Case2:
add $v0, $s3, $zero
For3:
bgt $v0, $s1, Exit
add $v1, $s0, $zero
For4: bgt $v1, $s2, EndFor4
beq $v0, $s1, InVien2
beq $v0, $s3, InVien2
beq $v1, $s0, InVien2
beq $v1, $s2, InVien2
sll $t8, $v0, 6
add $t8, $t8, $v1
sll $t8, $t8, 2
li $a1, GREEN
add $a2, $k0, $t8
```

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For4

InVien2:

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, RED

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For4

EndFor4:

add \$v0, \$v0, 1

j For3

Case3:

beq \$t1, 0, Case4

add \$v0, \$s1, \$zero

For5:

bgt \$v0, \$s3, Exit

add \$v1, \$s2, \$zero

For6:

bgt \$v1, \$s0, EndFor6

beq \$v0, \$s1, InVien3

beq \$v0, \$s3, InVien3

beq \$v1, \$s0, InVien3

beq \$v1, \$s2, InVien3

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, GREEN

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For6

InVien3:

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, RED

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For6

EndFor6:

add \$v0, \$v0, 1

j For5

Case4:

add \$v0, \$s3, \$zero

For7:

bgt \$v0, \$s1, Exit

add \$v1, \$s2, \$zero

For8:

bgt \$v1, \$s0, EndFor8

beq \$v0, \$s1, InVien4

beq \$v0, \$s3, InVien4

beq \$v1, \$s0, InVien4

beq \$v1, \$s2, InVien4

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, GREEN

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For8

InVien4:

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, RED

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For8

EndFor8:

add \$v0, \$v0, 1

j For7

Exit: li \$v0, 10

Syscall

Giả sử khi nhập tọa độ (x1,y1) và (x2,y2) lần lượt là (1;2), (40;60) kết quả thu được là:

