

Báo cáo Thực hành KTMT buổi 10

Họ và tên: Nguyễn Đức Phú

MSSV: 20215116

Assignment 1: Hiển thị 2 số cuối MSSV (16)

- Code:

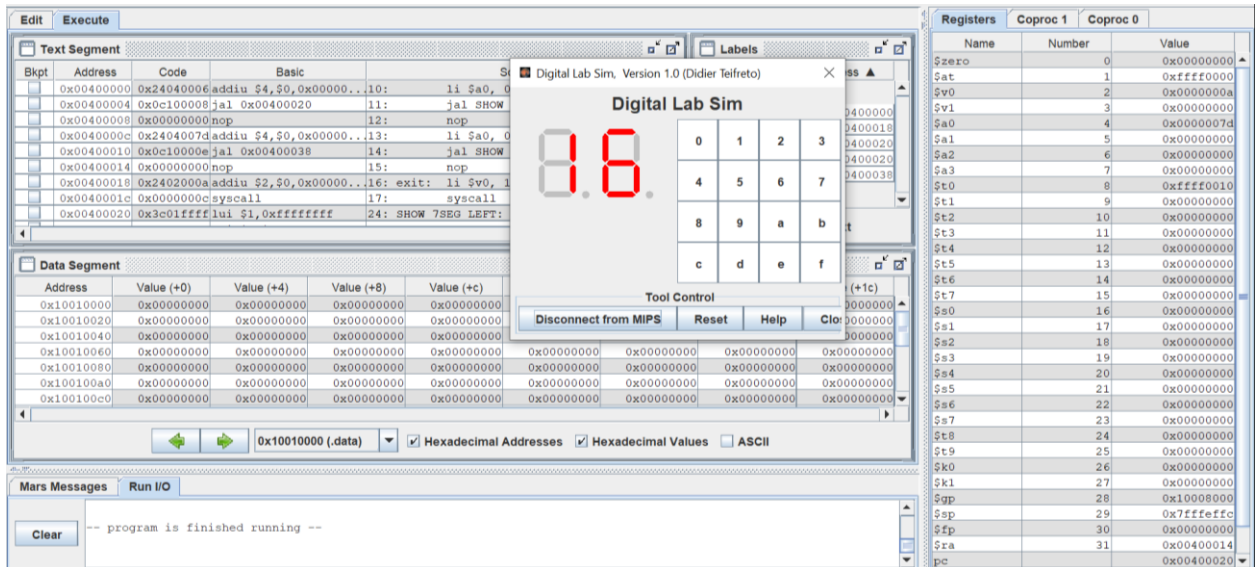
```
.eqv SEVENSEG_LEFT 0xFFFF0011
.eqv SEVENSEG_RIGHT 0xFFFF0010
.text
main:
    #MSSV:20215116
    #2 so cuoi cung: 16
    li $a0, 0x06
    jal SHOW_7SEG_LEFT          # show
    nop
    li $a0, 0x7D
    jal SHOW_7SEG_RIGHT        # show
    nop
exit:  li $v0, 10
      syscall
endmain:
#-----
# Function SHOW_7SEG_LEFT : turn on/off the 7seg
# param[in] $a0 value to shown
# remark $t0 changed
#-----
SHOW_7SEG_LEFT:  li    $t0, SEVENSEG_LEFT
                  sb    $a0, 0($t0)
                  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
                  sb    $a0, 0($t0) # assign new value
                  nop
                  jr    $ra
                  nop

```

- Kết quả chạy thử:



Assignment 2: Nhập vào một số nguyên từ bàn phím, hiển thị 2 chữ số cuối của số nguyên đó

- Code:


```

.eqv SEVENSEG_LEFT    0xFFFF0011
.eqv SEVENSEG_RIGHT   0xFFFF0010
.data
message: .asciiz "Nhap vao mot so nguyen: "
arr: .word 0x3F,0x6,0x5B,0x4F,0x66,0x6D,0x7D,0x7,0x7F,0x6F
#Ma de hien thi tu 0->9
.text
main:
    li    $v0, 4
    la    $a0, message
    syscall
    li    $v0, 5

```

```

    syscall
    move    $s0, $v0        #s0 = So nhap vao

    li     $t2, 10
    div    $s0, $s0, $t2
    mfhi $t1                # lay phan du khi chia 10
    sll    $t1, $t1, 2
    la     $a1, arr
    add    $a1, $a1, $t1
    lw     $a0, 0($a1)      #lay ra ma hien thi
    jal    SHOW_7SEG_RIGHT

#lam lai tuong tu hien thi so con lai
    la     $a1, arr
    div    $s0, $s0, $t2
    mfhi $t1
    sll    $t1, $t1, 2
    la     $a1, arr
    add    $a1, $a1, $t1
    lw     $a0, 0($a1)
    jal    SHOW_7SEG_LEFT

    li     $v0, 10          #Exit
    syscall

#-----
# 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
    sb     $a0, 0($t0)
    jr     $ra

#-----
# 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
    sb    $a0,    0($t0)
    jr    $ra
```

- Kết quả chạy thử khi nhập vào số 1234 và 9:

Digital Lab Sim, Version 1.0 (Didier Teireto)

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x24020004	addiu \$2,\$0,...	9:	li ...
0x00400004	0x3c011001	lui \$1,0x000...	10:	la ...
0x00400008	0x34240000	ori \$4,\$1,0x...		
0x0040000c	0x0000000c	syscall	11:	syscall
0x00400010	0x24020005	addiu \$2,\$0,...	12:	li ...
0x00400014	0x0000000c	syscall	13:	syscall
0x00400018	0x00028021	addu \$16,\$0,\$2	14:	move ...
0x0040001c	0x240a000a	addiu \$10,\$0,...	16:	li ...
0x00400020	0x15400001	bne \$10,\$0,0...	17:	div ...

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)
0x10010000	0x7061684e	0x6f617620	0x746f6d20	0x206f7320
0x10010020	0x00000006	0x0000005b	0x0000004f	0x000000e6
0x10010040	0x0000006f	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages

```
Nhap vao mot so nguyen: 1234
-- program is finished running --
```

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0xfffff000
\$v0	2	0x0000000a
\$v1	3	0x00000000
\$a0	4	0x0000004f
\$a1	5	0x10010028
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0xfffff011
\$t1	9	0x0000000c
\$t2	10	0x0000000a
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffcfc
\$fp	30	0x00000000
\$ra	31	0x00400080
pc		0x00400080

Digital Lab Sim, Version 1.0 (Didier Teireto)

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x24020004	addiu \$2,\$0,...	9:	li ...
0x00400004	0x3c011001	lui \$1,0x000...	10:	la ...
0x00400008	0x34240000	ori \$4,\$1,0x...		
0x0040000c	0x0000000c	syscall	11:	syscall
0x00400010	0x24020005	addiu \$2,\$0,...	12:	li ...
0x00400014	0x0000000c	syscall	13:	syscall
0x00400018	0x00028021	addu \$16,\$0,\$2	14:	move ...
0x0040001c	0x240a000a	addiu \$10,\$0,...	16:	li ...
0x00400020	0x15400001	bne \$10,\$0,0...	17:	div ...

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)
0x10010000	0x7061684e	0x6f617620	0x746f6d20	0x206f7320
0x10010020	0x00000006	0x0000005b	0x0000004f	0x000000e6
0x10010040	0x0000006f	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages

```
Nhap vao mot so nguyen: 9
-- program is finished running --
```

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0xfffff000
\$v0	2	0x0000000a
\$v1	3	0x00000000
\$a0	4	0x0000003f
\$a1	5	0x1001001c
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0xfffff011
\$t1	9	0x00000000
\$t2	10	0x0000000a
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffcfc
\$fp	30	0x00000000
\$ra	31	0x00400080
pc		0x00400080

Assignment 3:

- Code:

```
.eqv SEVENSEG_LEFT      0xFFFF0011
.eqv SEVENSEG_RIGHT     0xFFFF0010
.data
    message: .asciiz "Nhap vao mot ky tu: "
    arr: .word 0x3F,0x6,0x5B,0x4F,0x66,0x6D,0x7D,0x7,0x7F,0x6F
    #Ma de hien thi tu 0->9
.text
main:
    li    $v0, 4
    la    $a0, message
    syscall

    li    $v0, 12
    syscall

    move $s0, $v0

    li    $t2, 10
    div   $s0, $s0, $t2
    mfhi $t1          # lay phan du khi chia 10
    sll   $t1,$t1,2
    la    $a1, arr
    add   $a1,$a1,$t1
    lw    $a0,0($a1)   #lay ra ma hien thi
    jal   SHOW_7SEG_RIGHT

#lam lai tuong tu hien thi so con lai
    la    $a1, arr
    div   $s0, $s0, $t2
    mfhi $t1
    sll   $t1,$t1,2
    la    $a1, arr
    add   $a1,$a1,$t1
    lw    $a0,0($a1)
    jal   SHOW_7SEG_LEFT
```

```

        li    $v0, 10        #Exit
        syscall

#-----
# 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
        sb    $a0, 0($t0)
        jr    $ra

#-----
# 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
        sb    $a0, 0($t0)
        jr    $ra

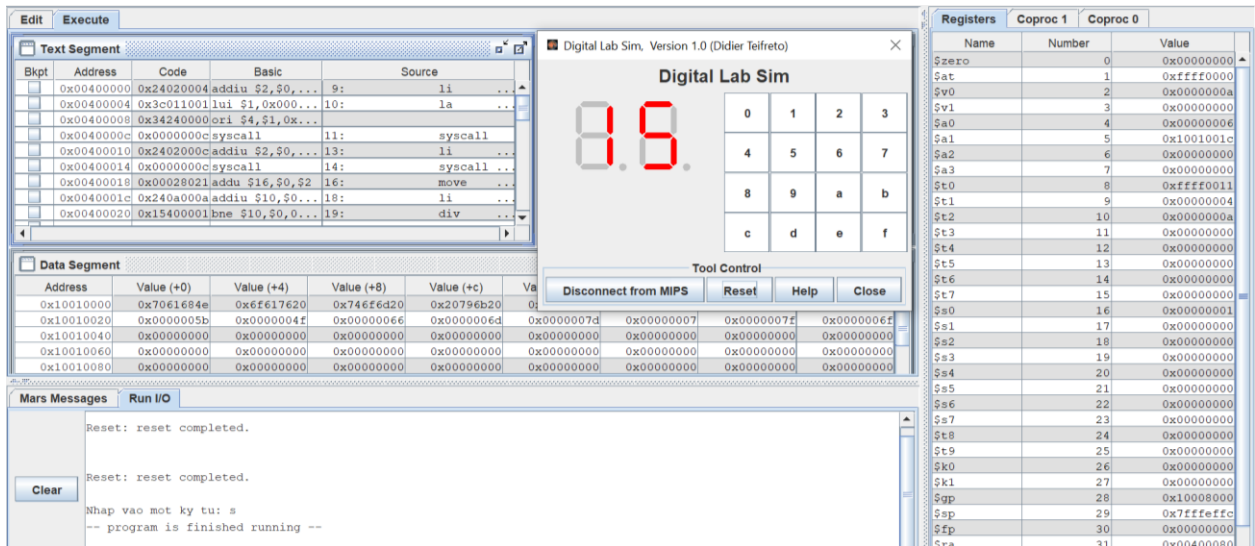
```

- Kết quả chạy thử:

Nhập ký tự A có mã là 65 và s có mã là 115

The screenshot shows the Digital Lab Sim software interface. On the left, the 'Text Segment' window displays assembly code with addresses, codes, and sources. Below it, the 'Data Segment' window shows memory addresses and values. At the bottom left, the 'Mars Messages' window shows the reset completion message and the input 'A'. In the center, the 'Digital Lab Sim' window features a 7-segment display showing '85' and a numeric keypad. On the right, the 'Registers' window lists registers \$zero through \$ra with their current values.

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0xffff0000
\$v0	2	0x0000000a
\$v1	3	0x00000000
\$a0	4	0x0000007d
\$a1	5	0x10010030
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0xffff0011
\$t1	9	0x00000018
\$t2	10	0x0000000a
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffcfc
\$fp	30	0x00000000
\$ra	31	0x00400080
pc		0x00400088



Assignment 4:

- Code:


```
.eqv MONITOR_SCREEN 0x10010000
.eqv RED 0x00FF0000
.eqv GREEN 0x0000FF00
.eqv BLUE 0x000000FF
.eqv WHITE 0x00FFFFFF
.eqv YELLOW 0x00FFFF00
.text
    li    $k0, MONITOR_SCREEN
    li    $t2, 2                # cac khoi tao
    li    $t4, 8
    li    $t1, -1
    li    $t4, 0
    li    $s1, 4
    add   $k0, $k0, -4

FOR:
    addi  $t1, $t1, 1           # i = 0, i ++
    addi  $t4, $t4, 1           # j++
    beq   $t1, 72, EXIT         # i = 72 stop
    add   $k0, $k0, $s1         # $k0 += 4
    div   $t1, $t2              # i / 2
```

```

mfhi $t3                # $t3 = i % 2

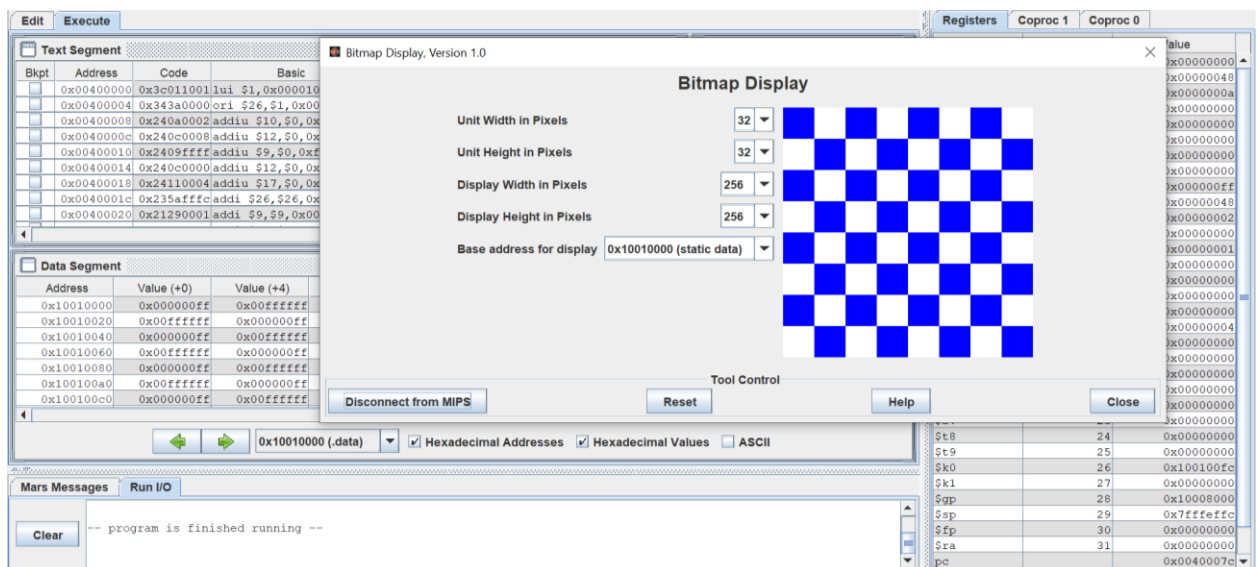
bne $t4, 8, continue # j = 8 => i++
li  $t4, 0              # j = 0
addi $t1, $t1, 1        # i++
continue:
    beq $t3, $zero, doi_mau
    li  $t0, WHITE
    sw  $t0, 0($k0)
    nop
    j   FOR

doi_mau:
    li  $t0, BLUE
    sw  $t0, 0($k0)
    nop
    j   FOR

EXIT:    li  $v0, 10
         syscall

```

- Kết quả chạy thử:



Assignment 5:

- Code:

```
.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 "x2 phai khac x1\n"
    error2: .asciiz "y2 phai khac y1\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     continue

Error1:
    li    $v0, 4
    la    $a0, error1
    syscall
    j     NhapX2
Error2:
    li    $v0, 4
    la    $a0, error2
    syscall
    j     NhapY2
continue:
    slt $t0, $s0, $s2
    slt $t1, $s1, $s3

    beq $t0, 0, Case3
    beq $t1, 0, Case2
Case1:  add $v0, $s1, $zero
loop_1:
    bgt $v0, $s3, Exit
    add $v1, $s0, $zero
loop_2:
    bgt $v1, $s2, endloop_2
    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   loop_2
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   loop_2
endloop_2:
        add $v0, $v0, 1
        j   loop_1

Case2:
        add $v0, $s3, $zero
loop_3:
        bgt $v0, $s1, Exit
        add $v1, $s0, $zero
loop_4:
        bgt $v1, $s2, endloop_4
        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     loop_4
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     loop_4
endloop_4:
        add   $v0, $v0, 1
        j     loop_3
Case3:
        beq   $t1, 0, Case4
        add   $v0, $s1, $zero
loop_5:
        bgt   $v0, $s3, Exit
        add   $v1, $s2, $zero
loop_6:
        bgt   $v1, $s0, endloop_6
        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     loop_6
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   loop_6
endloop_6:
        add $v0, $v0, 1
        j   loop_5
Case4:   add $v0, $s3, $zero
loop_7:
        bgt $v0, $s1, Exit
        add $v1, $s2, $zero
loop_8:
        bgt $v1, $s0, endloop_8
        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   loop_8
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   loop_8

```

```

endloop_8:
    add $v0, $v0, 1
    j    loop_7

```

```

Exit:    li    $v0, 10
        syscall

```

- Kết quả chạy thử:
Bộ số (3,27) và (27,3)

The screenshot shows a MIPS simulator interface. The main window displays assembly code with columns for Bkpt, Address, Code, and Basic. Below the code, there are sections for Text Segment and Data Segment. The Data Segment shows memory addresses and their corresponding values. On the right, a 'Bitmap Display' window is open, showing a green square on a black background, representing the output of the program. The display window has controls for Unit Width in Pixels, Unit Height in Pixels, Display Width in Pixels, and Display Height in Pixels. At the bottom, there is a 'Mars Messages' window showing the program's execution status and a 'Run I/O' button.

Bkpt	Address	Code	Basic
	0x00400000	0x3c011001	lui \$1,0x00001001
	0x00400004	0x343a0000	ori \$26,\$1,0x00000000
	0x00400008	0x24020004	addiu \$2,\$0,0x00000000
	0x0040000c	0x3c011001	lui \$1,0x00001001
	0x00400010	0x34240000	ori \$4,\$1,0x00000000
	0x00400014	0x0000000c	syscall
	0x00400018	0x24020005	addiu \$2,\$0,0x00000000
	0x0040001c	0x0000000c	syscall
	0x00400020	0x00028021	addu \$16,\$0,\$2

Address	Value (+0)	Value (+4)	Value (+8)
0x10010000	0x7061684e	0x3a317820	0x684e
0x10010020	0x79207061	0x00203a32	0x7020
0x10010040	0x61686b20	0x31792063	0x0000

Register	Value
\$s5	21
\$s6	22
\$s7	23
\$t8	24
\$t9	25
\$k0	26
\$k1	27
\$gp	28
\$sp	29
\$fp	30
\$ra	31
pc	0x004002a4

Mars Messages: -- program is finished running --