

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION
TECHNOLOGY



Báo cáo Bài tập lớn Cuối kỳ 20232

Học phần: Thực hành Kiến trúc máy tính Mã học phần: IT3280

TRẦN VĂN ĐỨC ANH
20225690

NGUYỄN BÁ ANH
20215260

Nhóm 15

Giảng viên hướng dẫn:

LÊ BÁ VUI

Hanoi, 12/06/2024

Nhiệm vụ

1. Nguyễn Bá Anh: Vẽ hình bằng ký tự ASCII
2. Trần Văn Đức Anh: Máy tính bỏ túi

Mục lục

1	Nguyễn Bá Anh: Vẽ hình bằng ký tự ASCII.....	2
1.1	Vấn đề.....	2
1.2	Thực hiện chương trình	3
1.3	Kết quả demo	4
1.4	Code	6
2	Trần Văn Đức Anh: Máy tính bỏ túi	16
2.1	Vấn đề.....	16
2.2	Hướng dẫn chạy chương trình	16
2.3	Phân tích cách thực hiện	17
2.4	Demo	17
2.4.1	Phép cộng.....	17
2.4.2	Phép trừ	19
2.4.3	Phép nhân	20
2.4.4	Phép chia và chia dư	21
2.4.5	Nhiều phép toán liên tiếp.....	24
2.5	Code	28

1 Nguyễn Bá Anh: Vẽ hình bằng ký tự ASCII

1.1 Vấn đề

Cho hình ảnh đã được chuyển thành các kí tự ASCII như hình vẽ. Đây là hình của chữ DCE có viền * và màu là các con số (1, 2, 3).

```

*****
*****
*2222222222222222*
*22222*****22222*
*22222*      *22222*
*22222*      *22222*      *****
*22222*      *22222*      **11111*****111*
*22222*      *22222*      **1111**      **
*22222*      *222222*      *1111*
*22222*****222222*      *11111*
*2222222222222222*      *11111*
*****
      ---
      /  o  o  \
      \   >  /
      -----
*****
*****
*33333333333333*
*33333*****
*33333*
*33333*****
*33333333333333*
*33333*****
*33333*
*33333*****
*33333333333333*
*****
*1111**
*1111****      *****
**111111***111*
*****
dce.hust.edu.vn

```

1. Hãy hiển thị hình ảnh trên lên giao diện console (hoặc giao diện Display trong công cụ giả lập Keyboard and Display MMIO Simulator)
2. Hãy sửa ảnh để các chữ cái DCE chỉ còn lại viền, không còn màu số ở giữa, và hiển thị
3. Hãy sửa ảnh để hoán đổi vị trí của các chữ, thành ECD, và hiển thị. Để đơn giản, các hoạt tiết đính kèm cũng được phép di chuyển theo.
4. Hãy nhập từ bàn phím kí tự màu cho chữ D, C, E, rồi hiển thị ảnh trên với màu mới.

Chú ý: Ngoài vùng nhớ lớn chứa ảnh được chứa sẵn trong code, không được tạo thêm vùng nhớ mới để chứa ảnh hiệu chỉnh.

1.2 Thực hiện chương trình

1. Định hướng

- B1: Ta chia hình ảnh trên thành các string theo từng dòng lưu vào bộ nhớ. Ví dụ ta sẽ có các string như sau:

```

String1: .asciiz "*****\n"
String2: "***** *3333333333333* \n"
String3: "*2222222222222222* *33333***** \n"
String4: "*22222*****22222* *33333* \n"
String5: "*22222* *22222* *33333***** \n"
String6: "*22222* *22222* ***** *3333333333333* \n"

```

```
String7: "*22222*   *22222*   **11111*****111* *33333***** \n"
String8: "*22222*   *22222*   **1111**   ** *33333*   \n"
String9: "*22222*   *22222*   *1111*   *33333***** \n"
String10:"*22222*****22222*   *11111*   *3333333333333* \n"
String11:"*2222222222222222*   *11111*   ***** \n"
String12:"*****   *11111*   \n"
String13 "   ---   *1111**   \n"
String14 "   / o o \ \   *1111****   *****   \n"
String15: \ \   > /   **111111***111*   \n"
String16 "   ----   *****   dce.hust.edu.vn   \n"
```

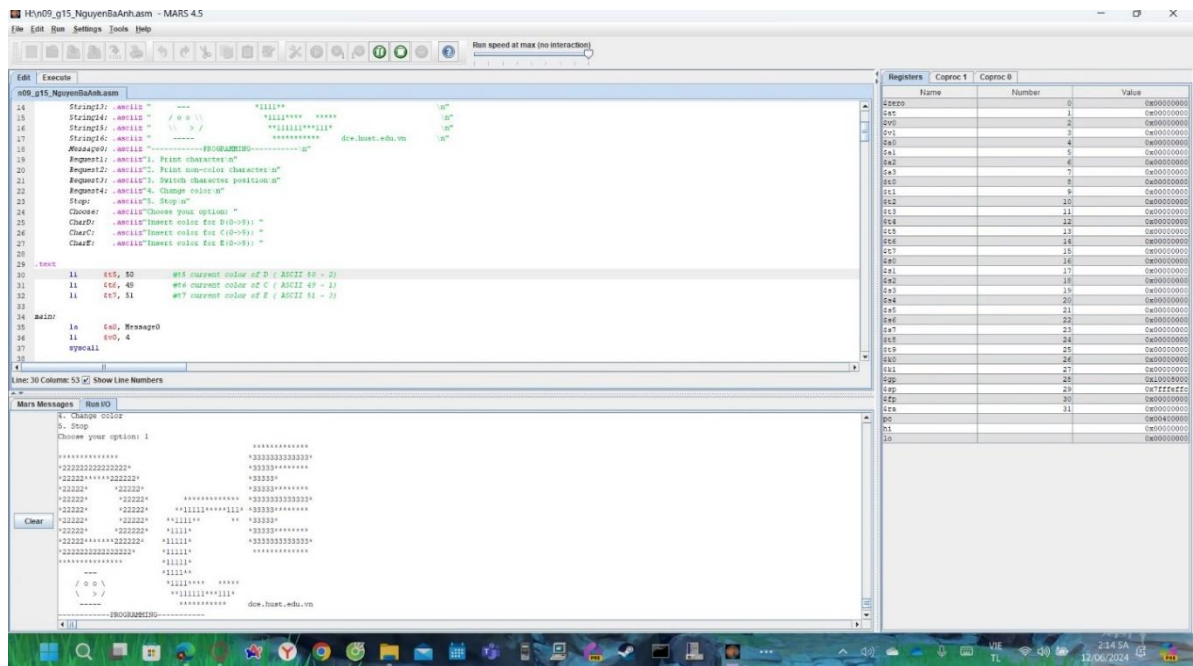
- B2. Với yêu cầu 1 ta tiến hành in từng dòng của hình ảnh.
- B3. Với yêu cầu 2 làm tương tự như yêu cầu 1 tuy nhiên khi in các kí tự của từng dòng ta sẽ kiểm tra xem kí tự đó có phải các chữ số từ 0-9 hay không nếu đúng ta sẽ tiến hành in khoảng trắng " ".
- B4. Với yêu cầu 3, ban đầu ta sẽ chia các string thành 3 phần của 3 chữ D, E, C Ví dụ ta sẽ chia string 5 nói trên thành 3 string nhỏ hơn. Sau đó ta sẽ in các string nhỏ ra theo thứ tự được yêu cầu từ đó sẽ được hình ảnh có vị trí các chữ cái bị đảo ngược.
- B5. Với yêu cầu 4, đầu tiên ta sẽ yêu cầu người dùng nhập các chữ số để biểu thị màu cho từng chữ cái. Sau đó ta tiến hành đổi màu các chữ cái theo từng dòng như sau:
 1. Duyệt qua từng kí tự của các string.
 2. Kiểm tra chữ cái đang được đổi màu là chữ cái nào. Ví dụ là chữ D.
 3. Tiến hành đổi màu khi gặp kí tự là các chữ số. Ví dụ thay 2 thành 3.
 4. Kiểm tra xem đã đi hết kí tự của chữ cái đang đổi màu chưa. Nếu chưa tiếp tục đổi màu chữ cái đó, rồi thì tiến hành đổi màu cho chữ cái khác.
 5. Lặp lại thao tác trên với các chữ cái còn lại

Khi hoàn thành đổi màu cho 1 dòng lập tức in dòng đó ra console. Làm tương tự như vậy cho đến khi hết string.

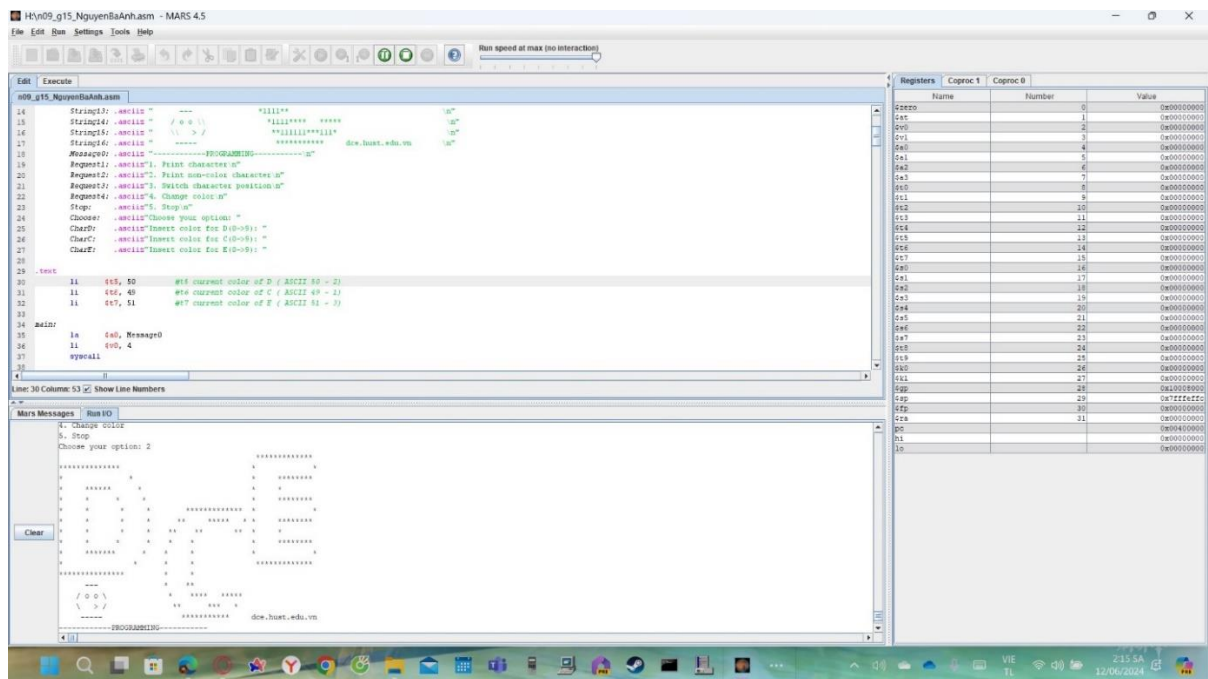
2. Kết quả chạy mã nguồn

1.3 Kết quả demo

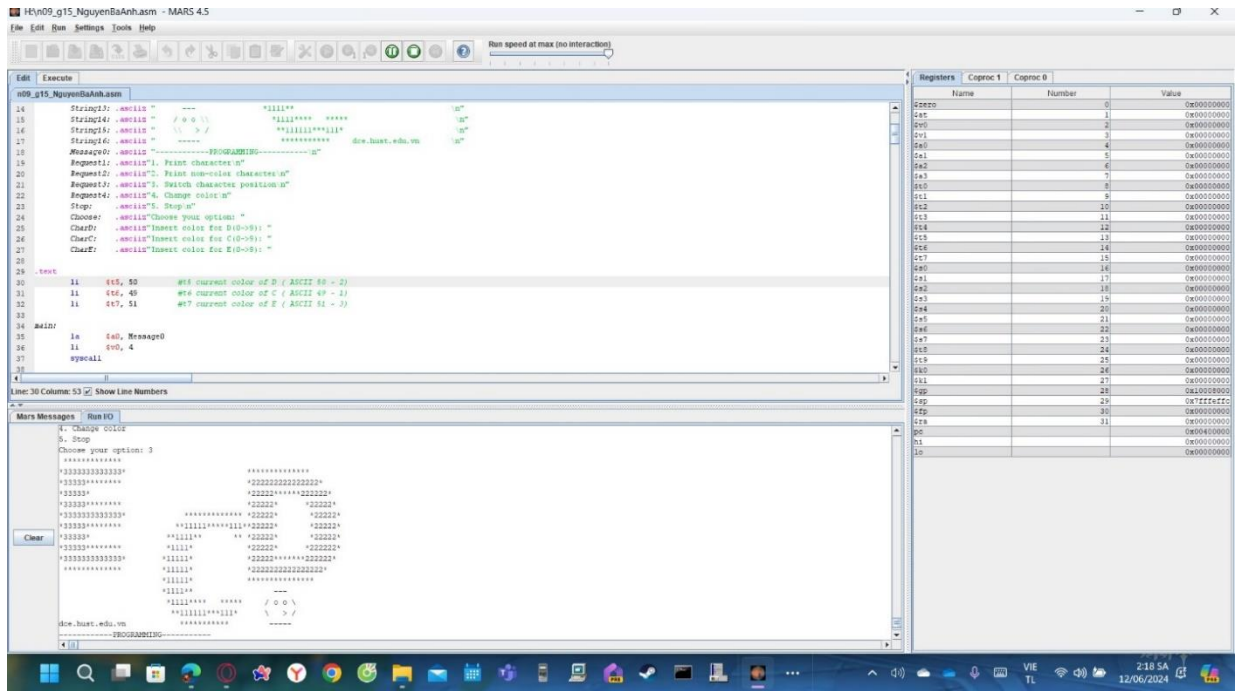
- Yêu cầu 1: in ra hình ảnh:



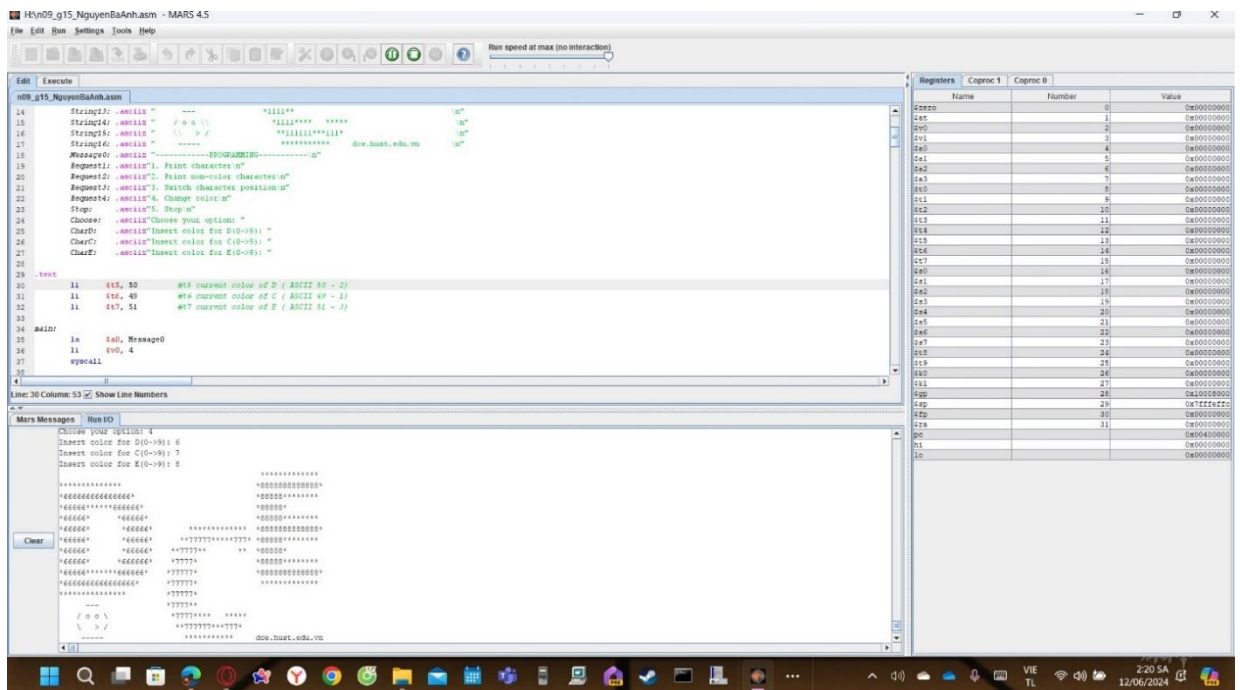
Yêu cầu 2: in ra ảnh không màu:



Yêu cầu 3: in ra hình ảnh bị đổi vị trí:



Yêu cầu 4: đổi màu cả 3 character:



1.4 Code

.data

```
String1: .asciiz "***** \n"
String2: .asciiz "***** *333333333333 \n"
String3: .asciiz "*222222222222* *3333***** \n"
```

```

String4: .asciiz "*22222*****22222*          *33333*          \n"
String5: .asciiz "*22222*      *22222*          *33333*****          \n"
String6: .asciiz "*22222*      *22222*      *****          *3333333333333* \n"
String7: .asciiz "*22222*      *22222*      **11111*****111* *33333***** \n"
String8: .asciiz "*22222*      *22222*      **1111**      ** *33333*          \n"
String9: .asciiz "*22222*      *22222*      *1111*          *33333*****          \n"
String10: .asciiz "*22222*****22222*      *11111*          *3333333333333* \n"
String11: .asciiz "*2222222222222222*      *11111*          *****          \n"
String12: .asciiz "*****          *11111*          \n"
String13: .asciiz "      ---          *1111**          \n"
String14: .asciiz "      / o o \\\          *1111****          *****          \n"
String15: .asciiz "      \\\ > /          **111111****111*          \n"
String16: .asciiz "      ----          *****          dce.hust.edu.vn          \n"
Message0: .asciiz "-----PROGRAMMING-----\n"
Request1: .asciiz"1. Print character\n"
Request2: .asciiz"2. Print non-color character\n"
Request3: .asciiz"3. Switch character position\n"
Request4: .asciiz"4. Change color\n"
Stop: .asciiz"5. Stop\n"
Choose: .asciiz"Choose your option: "
CharD: .asciiz"Insert color for D(0->9): "
CharC: .asciiz"Insert color for C(0->9): "
CharE: .asciiz"Insert color for E(0->9): "

```

.text

```

li      $t5, 50 #t5 current color of D ( ASCII 50 ~ 2)
li      $t6, 49 #t6 current color of C ( ASCII 49 ~ 1)
li      $t7, 51 #t7 current color of E ( ASCII 51 ~ 3)

```

main:

```

la      $a0, Message0
li      $v0, 4

```

syscall

la \$a0, Request1

li \$v0, 4

syscall

la \$a0, Request2

li \$v0, 4

syscall

la \$a0, Request3

li \$v0, 4

syscall

la \$a0, Request4

li \$v0, 4

syscall

la \$a0, Stop

li \$v0, 4

syscall

la \$a0, Choose

li \$v0, 4

syscall

li \$v0, 5

syscall

Case1menu:

addi \$v1, \$0, 1

bne \$v0, \$v1, Case2menu

j Menu1

Case2menu:

```
addi    $v1, $0, 2
bne     $v0, $v1, Case3menu
j       Menu2
```

Case3menu:

```
addi    $v1, $0, 3
bne     $v0, $v1, Case4menu
j       Menu3
```

Case4menu:

```
addi    $v1, $0, 4
bne     $v0, $v1, Case5menu
j       Menu4
```

Case5menu:

```
addi    $v1, $0, 5
bne     $v0, $v1, defaultmenu
j       Exit
```

defaultmenu:

j main

#Print character line by line

Menu1:

```
addi    $t0, $0, 0
addi    $t1, $0, 16
la      $a0, String1
```

Loop:

```

    beq    $t1, $t0, main
    li     $v0, 4
    syscall

```

```

    addi   $a0, $a0, 68
    addi   $t0, $t0, 1
    j      Loop

```

Menu2:

```

    addi   $s0, $0, 0
    addi   $s1, $0, 16
    la     $s2, String1

```

Loop1:

```

    beq    $s1, $s0, main
    addi   $t0, $0, 0
    addi   $t1, $0, 68

```

#Find number in line

PrintNonColor:

```

    beq    $t1, $t0, End
    lb     $t2, 0($s2)
    bgt    $t2, 47, Label
    j      Tmp

```

#Print " " if the character is number

Label:

```

    bgt    $t2, 57, Tmp
    addi   $t2, $0, 0x20
    j      Tmp

```

#Print the fixed line

Tmp:

```
li    $v0, 11
addi  $a0, $t2, 0
syscall

addi  $s2, $s2, 1
addi  $t0, $t0, 1
j     PrintNonColor
```

End:

```
addi  $s0, $s0, 1
j     Loop1
```

Menu3:

```
addi  $s0, $0, 0
addi  $s1, $0, 16
la     $s2, String1
```

Loop3:

```
beq    $s1, $s0, main
```

#Split each line into 3 section "D, E, C"

```
sb     $0, 21($s2)      #Set "\0" value to split the string
sb     $0, 43($s2)
sb     $0, 65($s2)
```

#Change string position

```
li     $v0, 4           #Print each line of E and it will stop when meet "\0" value
la     $a0, 44($s2)
syscall
```

```
li     $v0, 4           #Print each line of C
```

```
la    $a0, 22($s2)
```

```
syscall
```

```
li    $v0, 4                #Print each line of D
```

```
la    $a0, 0($s2)
```

```
syscall
```

```
li    $v0, 4                #Print newline character
```

```
la    $a0, 66($s2)
```

```
syscall
```

```
#Remove "\0" from string
```

```
addi  $t1, $0, 0x20
```

```
sb    $t1, 21($s2)
```

```
sb    $t1, 43($s2)
```

```
sb    $t1, 65($s2)
```

```
addi  $s0, $s0, 1
```

```
addi  $s2, $s2, 68
```

```
j     Loop3
```

Menu4:

```
#Change color for D
```

```
InsertColorD:
```

```
li    $v0, 4
```

```
la    $a0, CharD
```

```
syscall
```

```
li    $v0, 5
```

```
syscall
```

```

blt    $v0, 0, InsertColorD
bgt    $v0, 9, InsertColorD
addi   $s3, $v0, 48

```

#Change color for C

InsertColorC:

```

li      $v0, 4
la      $a0, CharC
syscall

```

```

li      $v0, 5
syscall

```

```

blt     $v0, 0, InsertColorC
bgt     $v0, 9, InsertColorC
addi    $s4, $v0, 48

```

#Change color for E

InsertColorE:

```

li      $v0, 4
la      $a0, CharE
syscall

```

```

li      $v0, 5
syscall

```

```

blt     $v0, 0, InsertColorE
bgt     $v0, 9, InsertColorE
addi    $s5, $v0, 48

```

```

addi    $s0, $0, 0
addi    $s1, $0, 16

```

```

la    $s2,String1
li    $a1, 48
li    $a2, 57

```

#Begin to change string color

ChangeColorLoop:

```

    beq    $s1, $s0, updateColor    #When $s1 = $s0 aka finish loop through 16 strings change the
value of "$t5,6,7" - color storing address

```

```

    addi    $t0, $0, 0
    addi    $t1, $0, 68

```

PrintChangeColor:

```

    beq    $t1, $t0, EndChangeColor    #When $t1 = $t0 aka finish loop through 68 chars of a string print
it

```

```

    lb     $t2, 0($s2)

```

CheckD:

```

    bgt    $t0, 21, CheckC    #If $t0 >= 21 aka $t0 store the index of the beginning of C switch to
CheckC

```

```

    beq    $t2, $t5, fixD    #If $t2 = $t5 aka the color was not changed jump to fixD to change color
    j      TmpChangeColor

```

CheckC:

```

    bgt    $t0, 43, CheckE
    beq    $t2, $t6, fixC
    j      TmpChangeColor

```

CheckE:

```

    beq    $t2, $t7, fixE
    j      TmpChangeColor

```

fixD:

```

    sb     $s3, 0($s2)

```

```
j      TmpChangeColor
```

fixC:

```
sb      $s4, 0($s2)
j      TmpChangeColor
```

fixE:

```
sb      $s5, 0($s2)
j      TmpChangeColor
```

#Increase the value of \$t0 and \$s2 to check the next character

TmpChangeColor:

```
addi    $s2, $s2, 1
addi    $t0, $t0, 1
j      PrintChangeColor
```

#Print string after change color

EndChangeColor:

```
li      $v0, 4
addi    $a0, $s2, -68
syscall

addi    $s0, $s0, 1
j      ChangeColorLoop
```

updateColor:

```
move    $t5, $s3
move    $t6, $s4
move    $t7, $s5
j      main
```

Exit:

2 Trần Văn Đức Anh: Máy tính bỏ túi

2.1 Vấn đề

Sử dụng 2 ngoại vi là bàn phím và LED 7-thanh để xây dựng một máy tính bỏ túi đơn giản. Hỗ trợ các phép toán cộng, trừ, nhân, chia, chia dư. Do trên bàn phím không có các phím trên nên sẽ dùng các phím:

1. Bấm phím a để nhập phép tính cộng.
2. Bấm phím b để nhập phép tính trừ.
3. Bấm phím c để nhập phép tính nhân.
4. Bấm phím d để nhập phép tính chia.
5. Bấm phím e để nhập phép chia dư.
6. Bấm phím f để nhập phép bằng.

Yêu cầu cụ thể như sau:

1. Khi nhấn các phím số, hiển thị lên LED, do chỉ có 2 LED nên chỉ hiển thị 2 số cuối cùng. Ví dụ khi nhấn phím 1 → hiển thị 01. Khi nhấn thêm phím 2 → hiển thị 12. Khi nhấn thêm phím 3 → hiển thị 23.
2. Sau khi nhập số, sẽ nhập phép tính cộng, trừ, nhân, chia, chia dư.
3. Sau khi nhấn phím f để tính toán và hiển thị kết quả trên LED 7-thanh.
4. Có thể thực hiện các phép toán liên tiếp.

2.2 Hướng dẫn chạy chương trình

1. Mở công cụ Digital Lab Sim, Connect to MIPS và chạy chương trình.
2. Nhập các số bằng cách nhập từ các phím trên công cụ Digital Lab Sim và màn hình led sẽ hiển thị 2 chữ số cuối.
3. Nhập xong số đầu tiên thì nhập toán tử cộng, trừ, nhân, chia tương ứng với a, b, c, d, e hiện trên phím của công cụ Digital Lab Sim.
4. Nhập số thứ hai.
5. Nhấn f (tương đương dấu '=') để hiển thị 2 chữ số cuối cùng của kết quả.

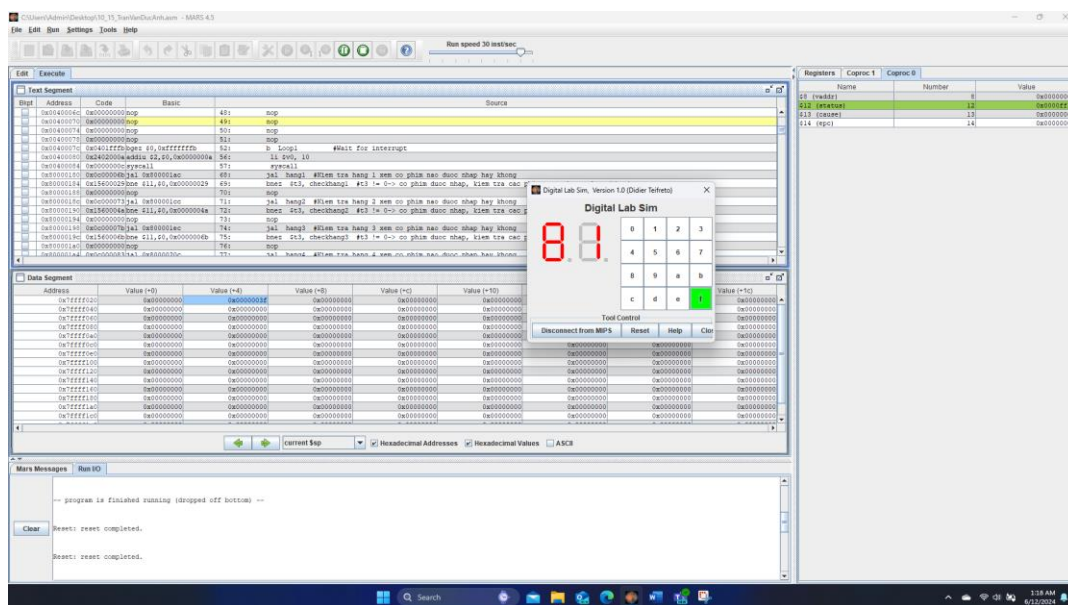
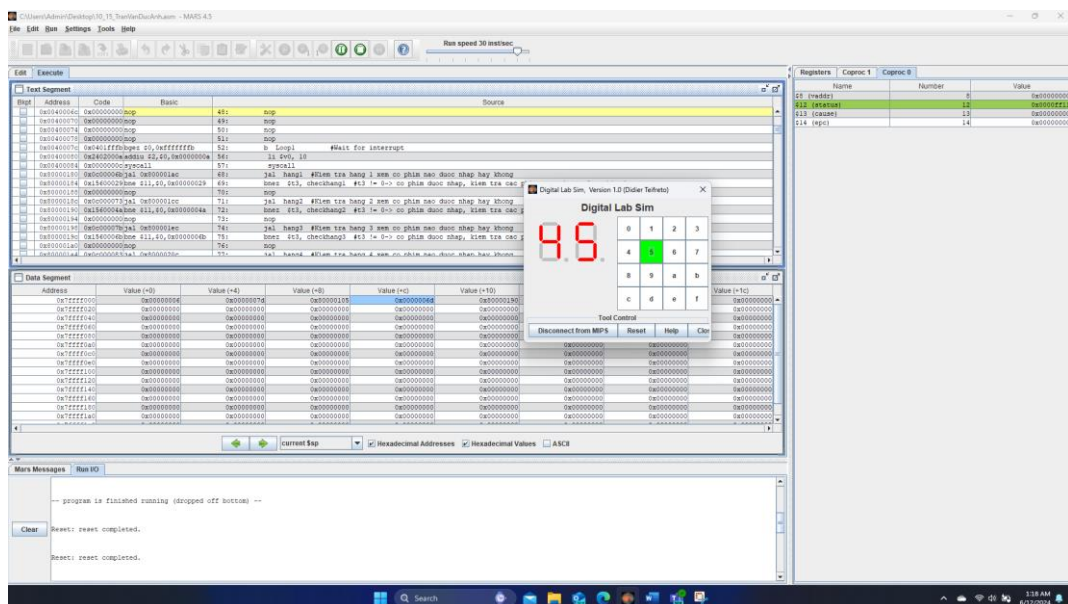
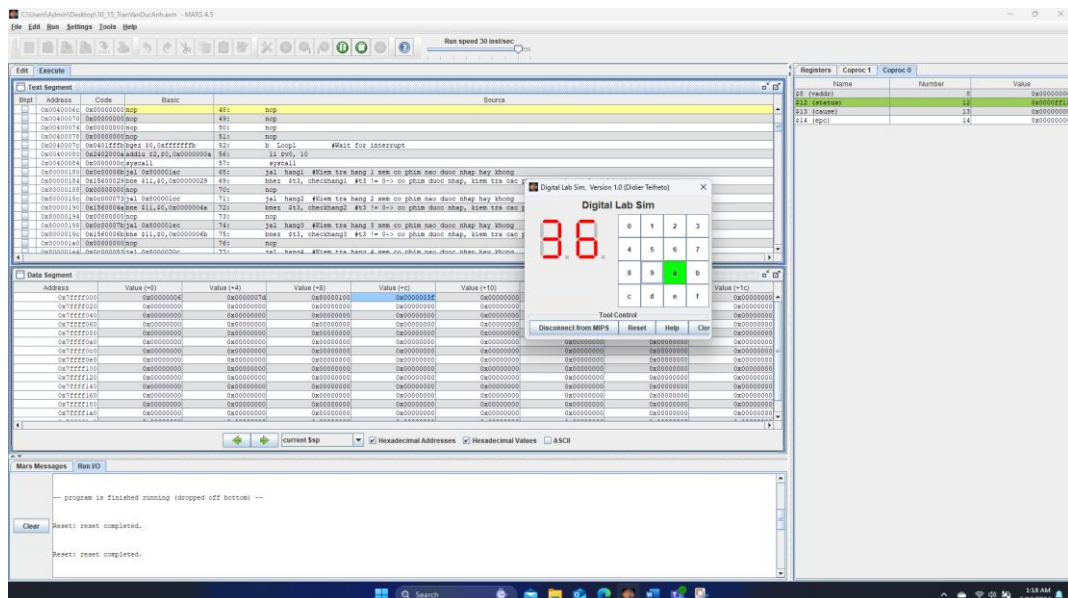
2.3 Phân tích cách thực hiện

1. Nhập từ bàn phím ảo của công cụ Digital Lab Sim và xác định mã của phím vừa nhận.
2. Thực hiện kiểm tra:
 - (a) Nếu đó là một số: update lại giá trị tạm thời, và giá trị của số đó sẽ xuất hiện trên led 7 đoạn của công cụ Digital Lab Sim.
 - (b) Nếu đó là một chữ (là một toán tử): Lưu lại toán tử đó (cộng, trừ, nhân, chia lấy phần nguyên, chia lấy phần dư). Lưu the giá trị số của số đầu tiên vào thanh ghi và chuyển sang nhập số thứ hai.
3. Nếu toán tử nhập vào là '=' (dấu bằng) -> .Lưu giá trị của số thứ 2, sau đó tính toán và hiển thị kết quả ra màn hình trên thanh led 7 đoạn của Digital Lab Sim.
4. Những lưu ý:
 - (a) Khi chọn toán tử là '/' (phép chia lấy phần nguyên) hoặc '(phép chia lấy phần dư) nếu số hạng thứ 2 bằng 0 thì in ra: 'Kết quả không tồn tại. Không thể chia cho số 0'. Đồng thời nhập lại số hạng thứ 2 để thực hiện phép tính.
 - (b) Nếu nhập 1 lúc 2 toán tử mà không nhập số hạng thứ 2 thì chương trình sẽ coi như số hạng thứ 2 bằng 0 và thực hiện phép tính số hạng thứ nhất với toán tử đầu tiên được nhập vào.
 - (c) Không thể nhập cùng một lúc 2 toán tử để tính toán (ví dụ: nhập dấu '*' sau đó nhập luôn dấu '-' thì kết quả sẽ ra như lưu ý thứ 2).
 - (d) Hiện tại thì phép tính chưa thể xử lí nhiều toán hạng cùng một lúc (một phép tính chỉ có thể có một toán tử và 2 số hạng).
 - (e) Trong trường hợp phép trừ, nếu số hạng thứ nhất nhỏ hơn số hạng thứ hai thì kết quả sẽ ra âm. Khi đó kết quả sẽ không hiển thị trên led 7 đoạn của công cụ Digital Lab Sim.

2.4 Demo

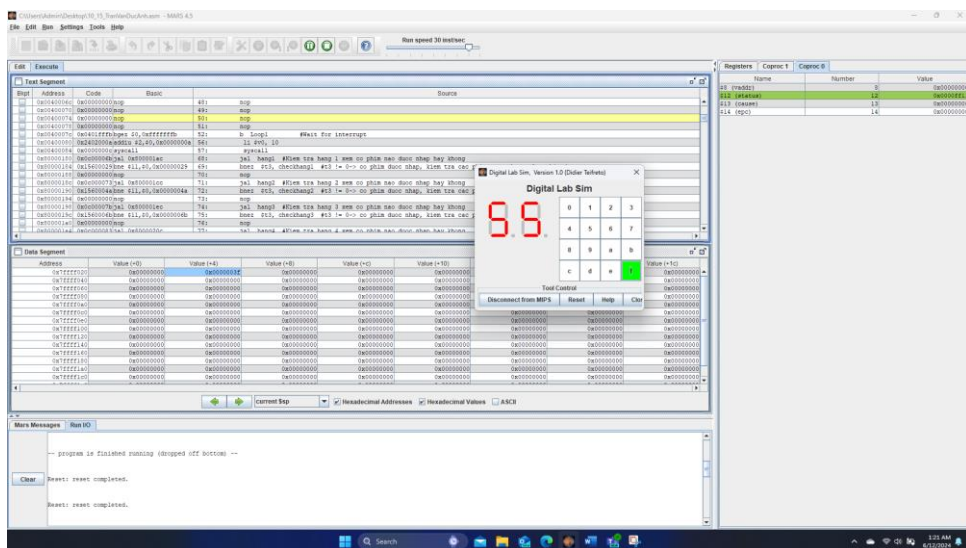
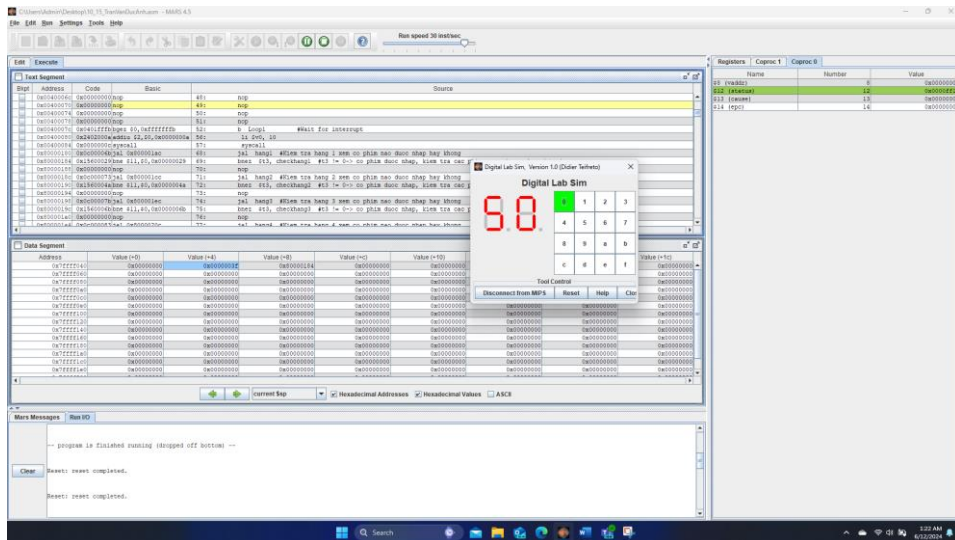
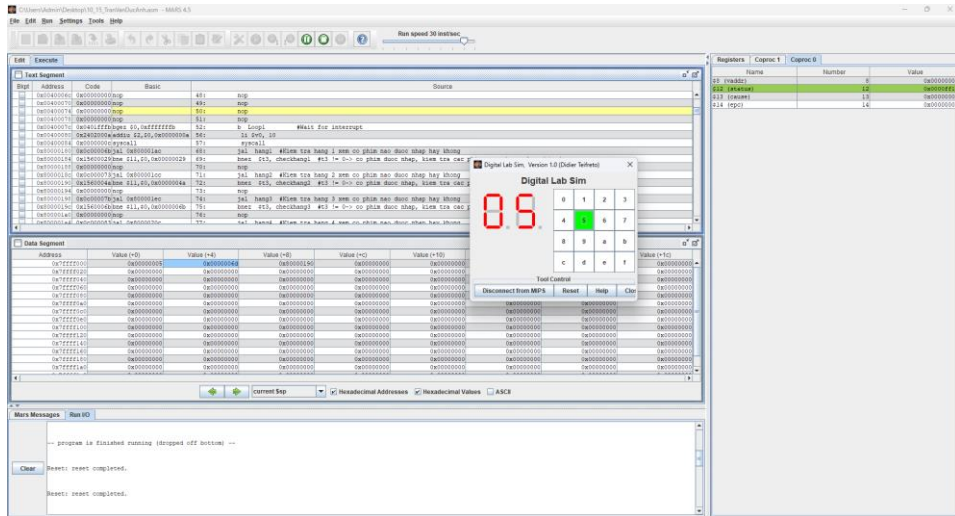
2.4.1 Phép cộng

$$36 + 45 = 81$$



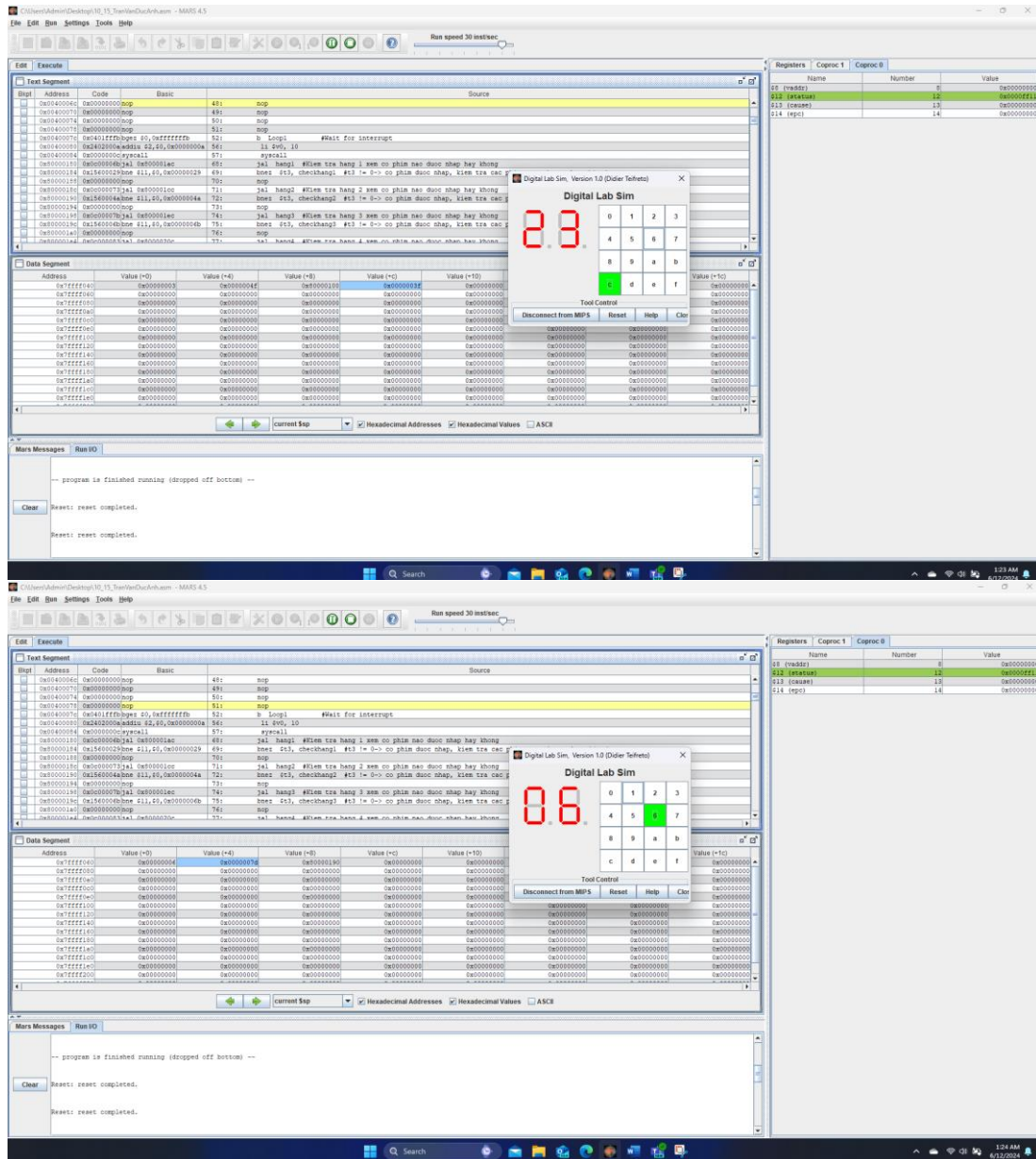
2.4.2 Phép trừ

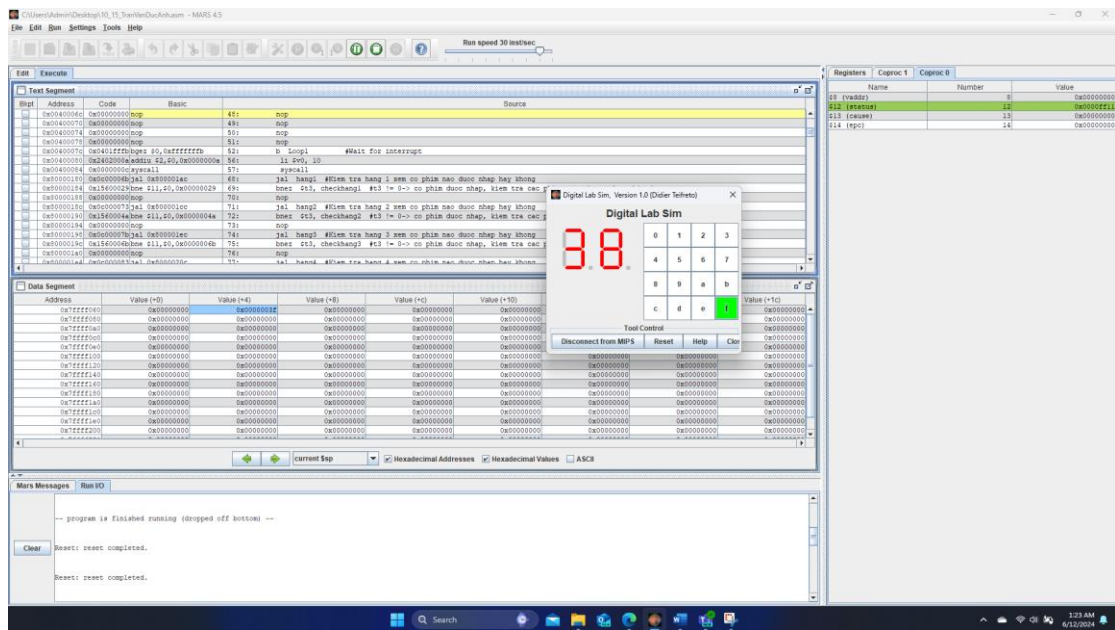
105 - 50 = 55



2.4.3 Phép nhân

$$123 * 6 = 738$$

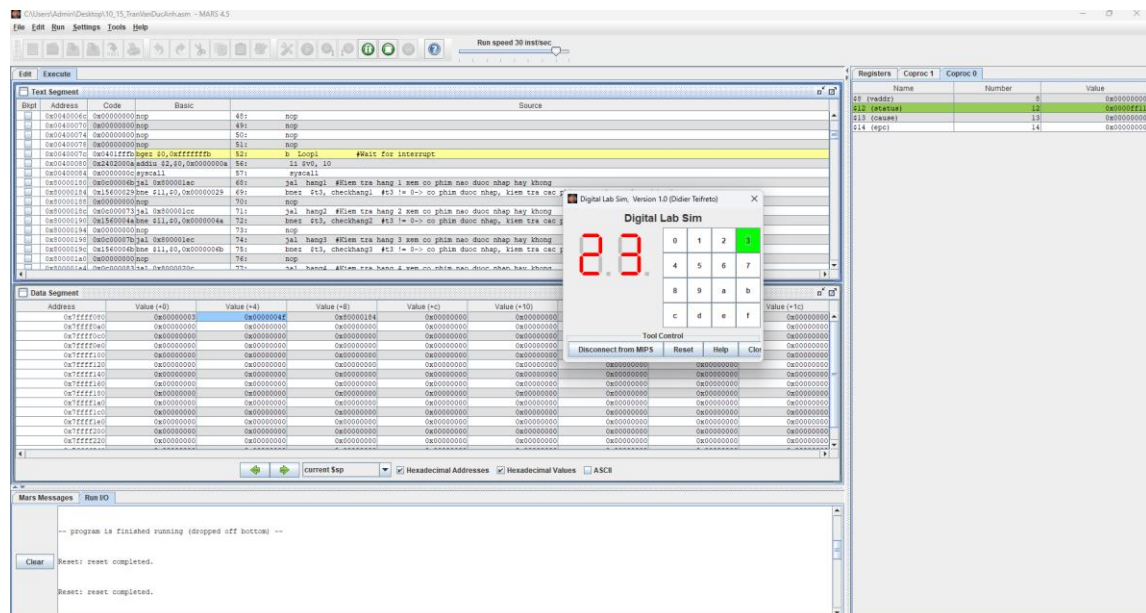


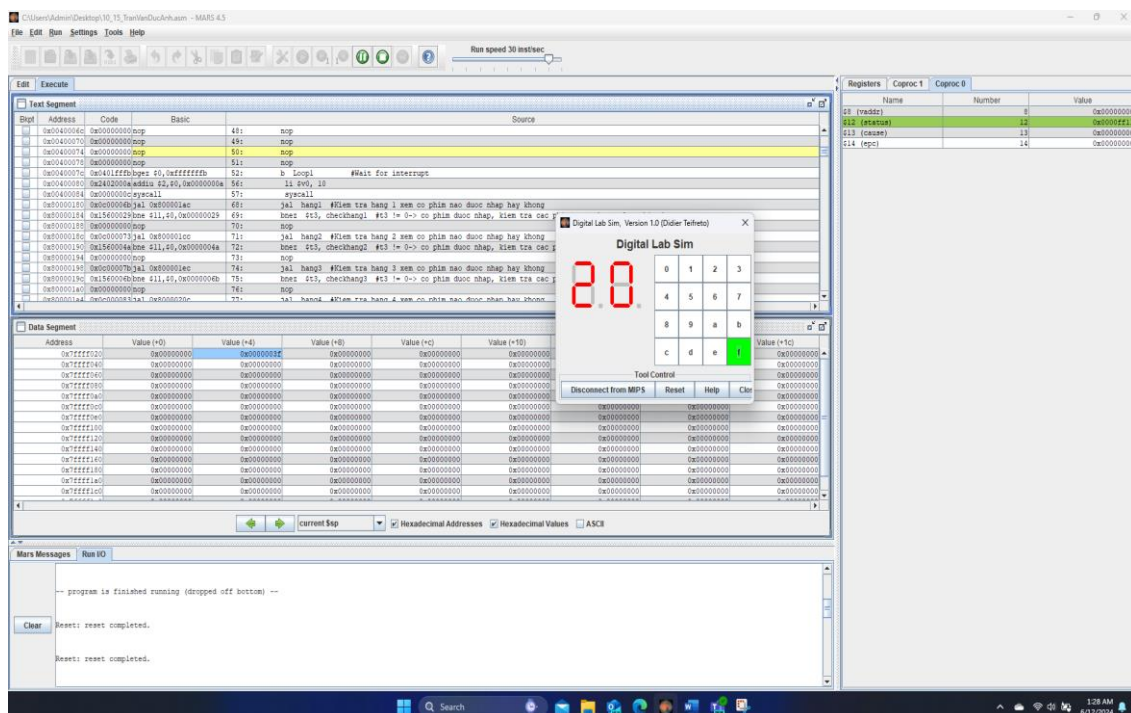
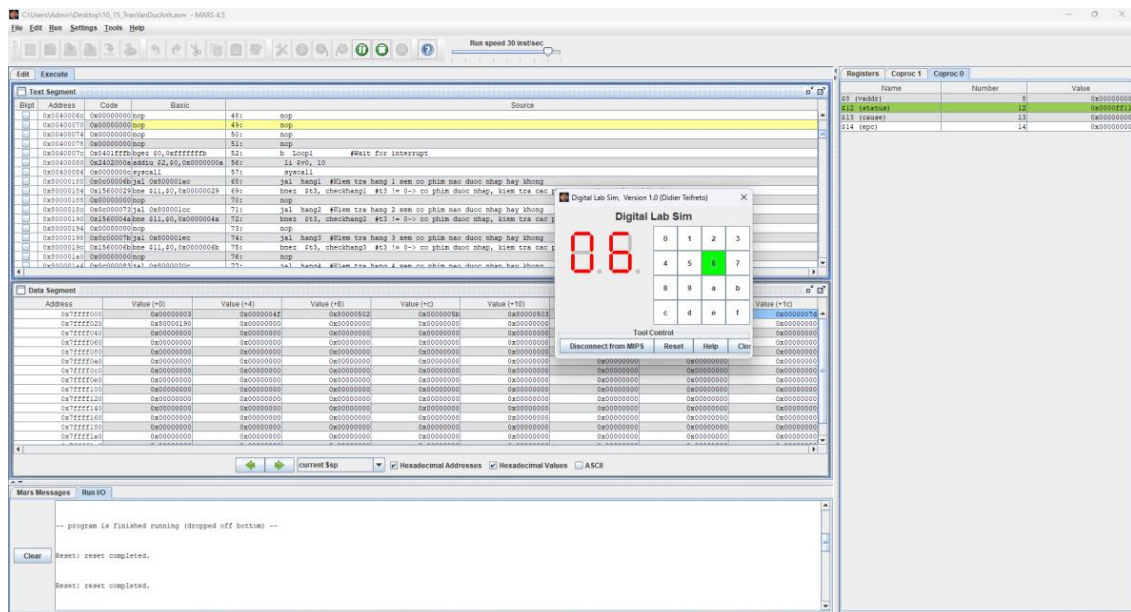


2.4.4 Phép chia và chia dư

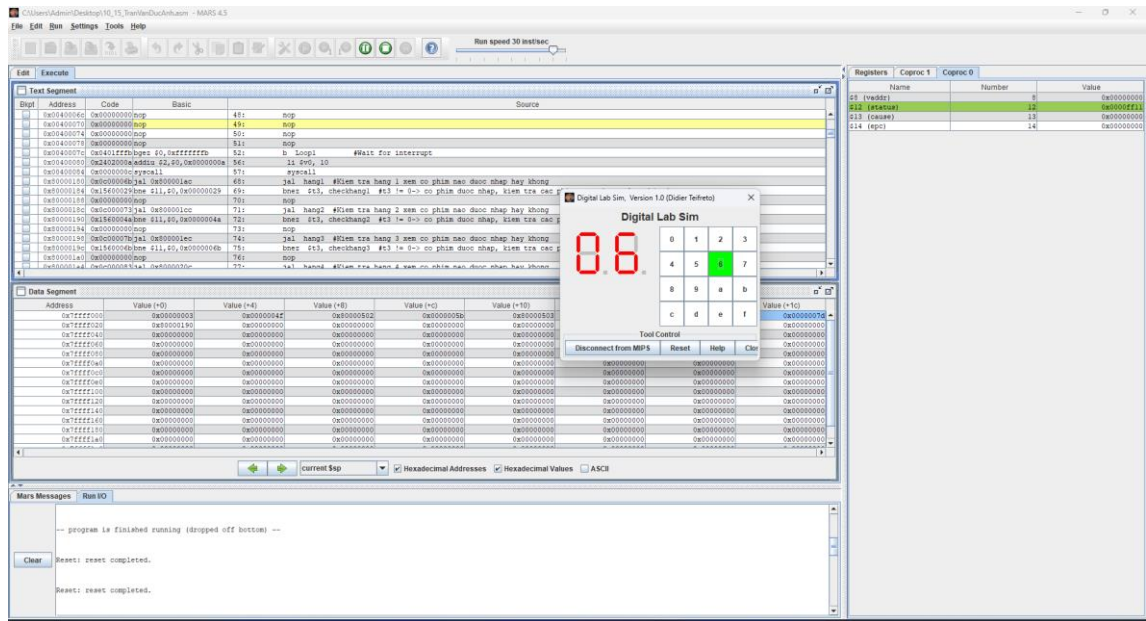
$$123 / 6 = 20$$

Phép chia thường





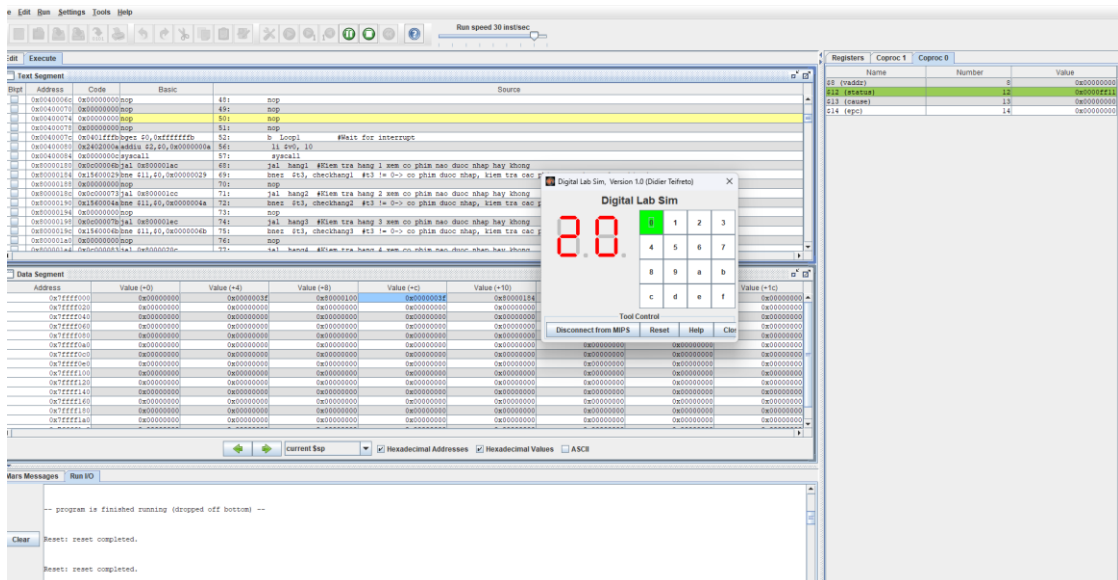
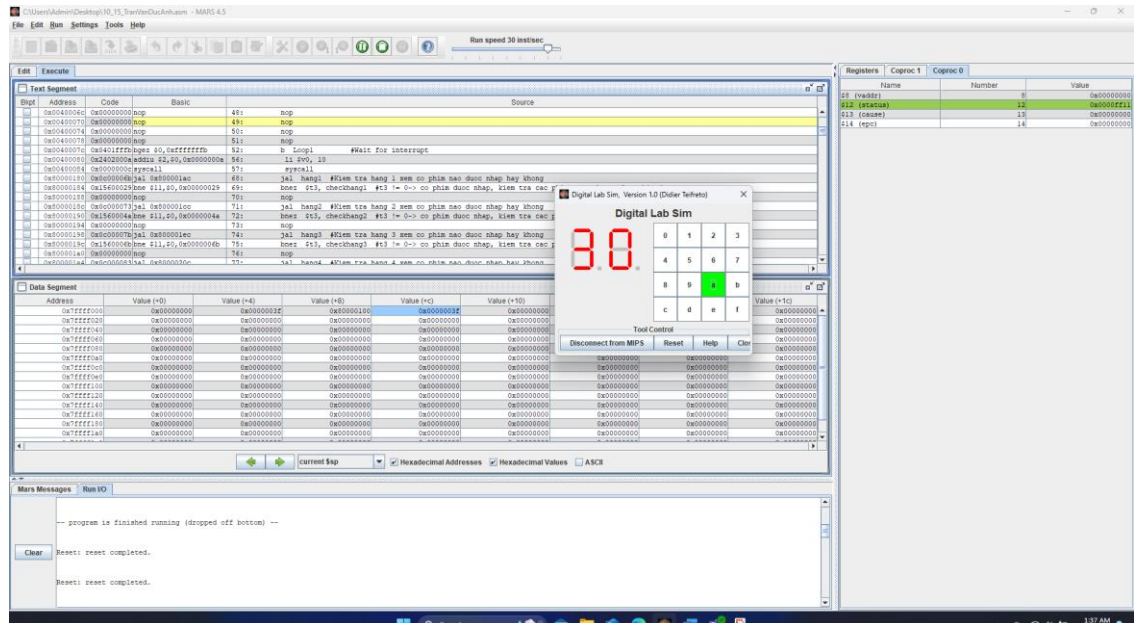
Phép chia dư $123\%6=3$

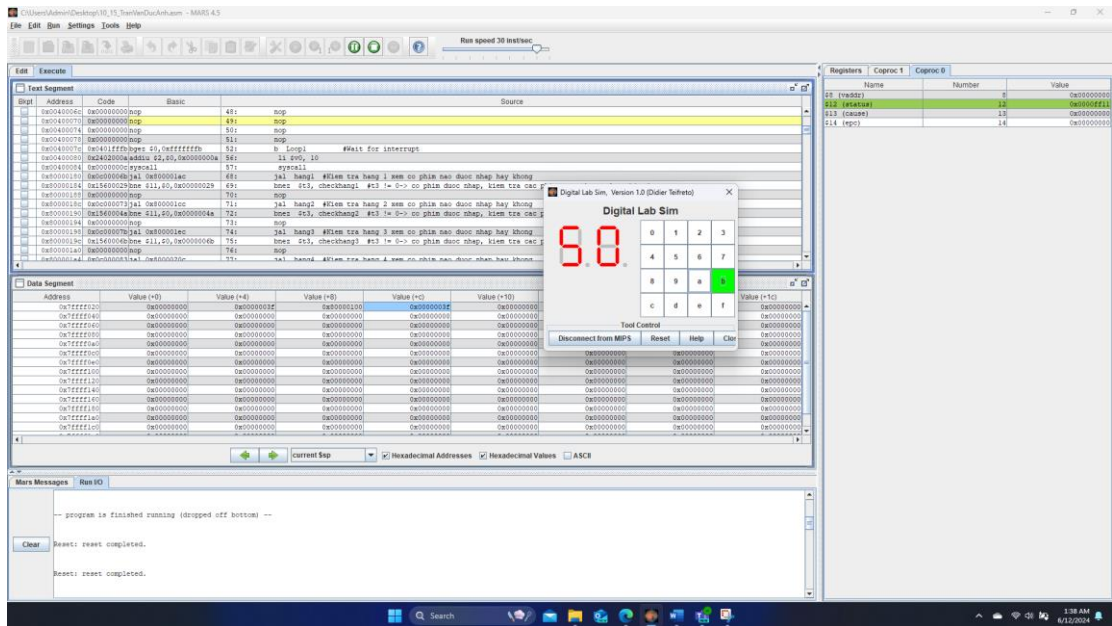


2.4.5 Nhiều phép toán liên tiếp

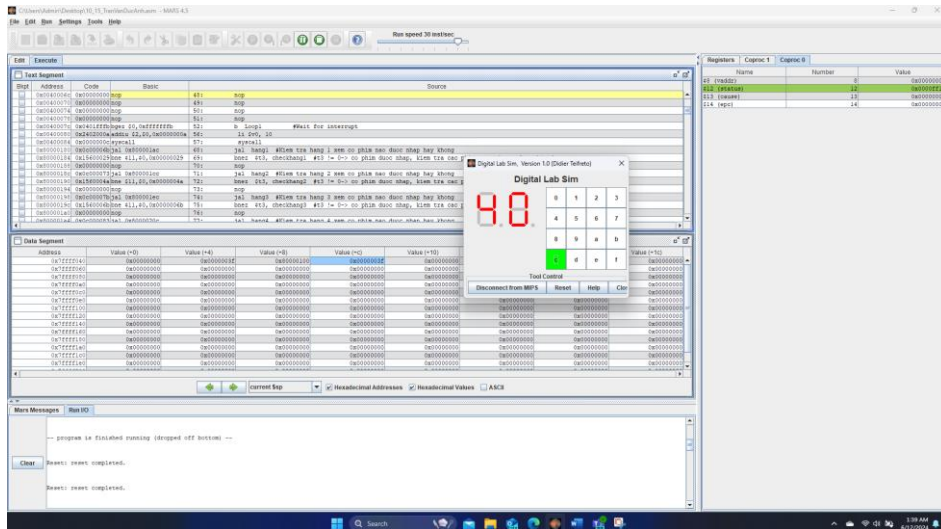
$$(30+20-10)*2/4\%3=2$$

$$30 + 20 = 50$$

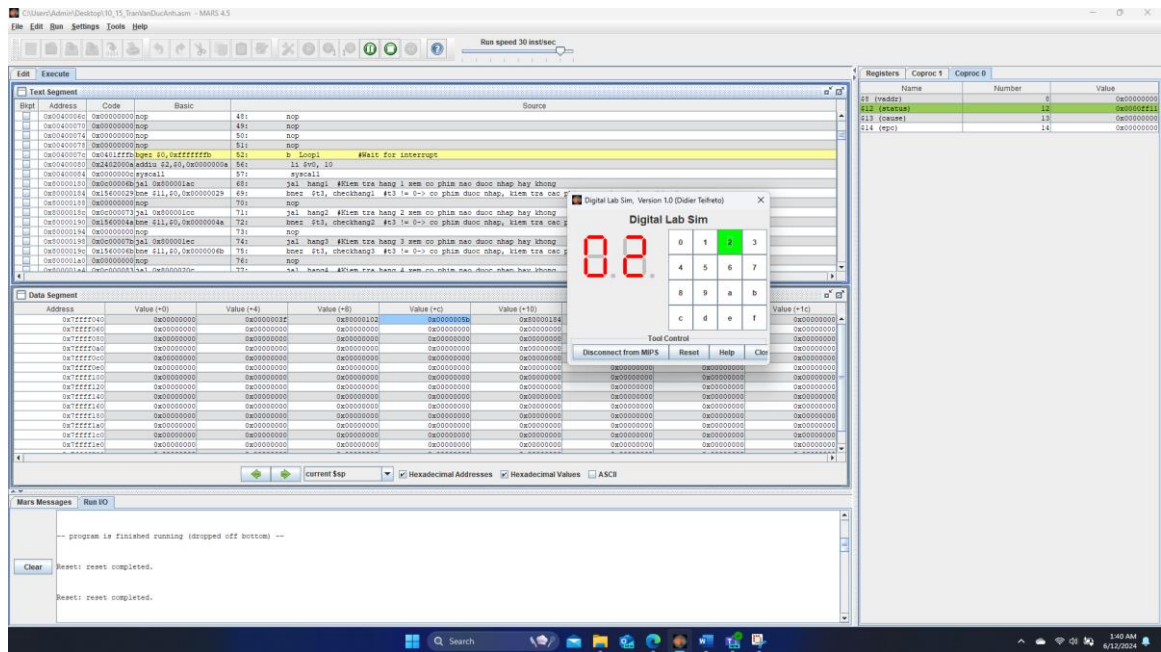




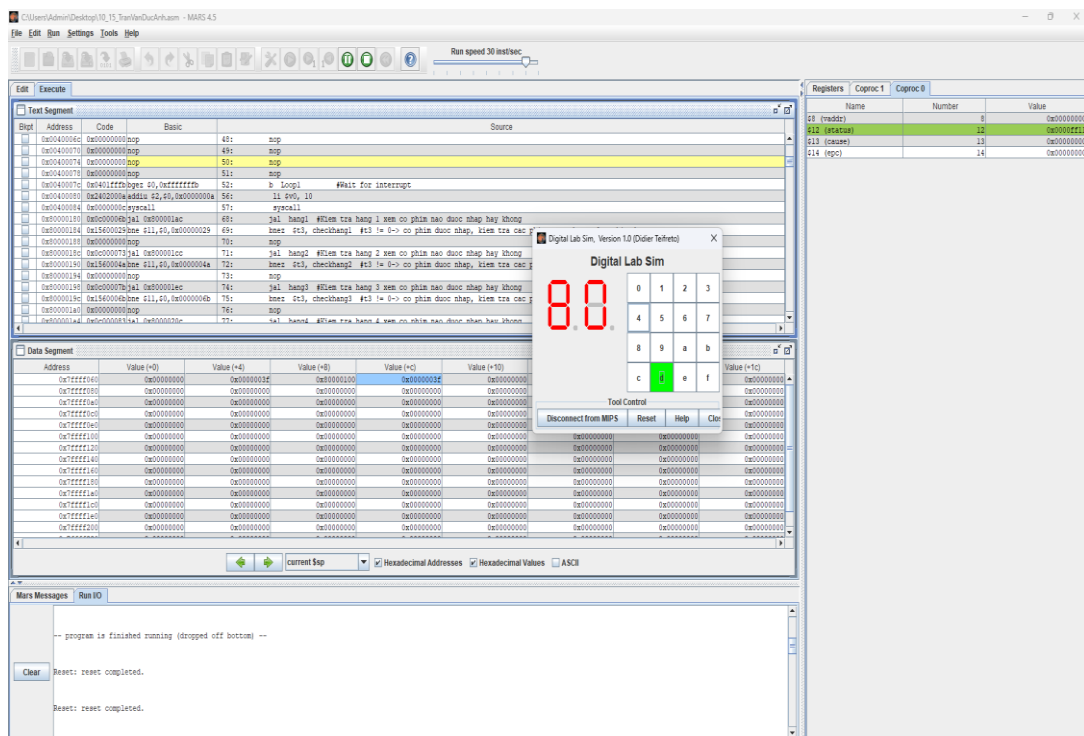
50-10=40

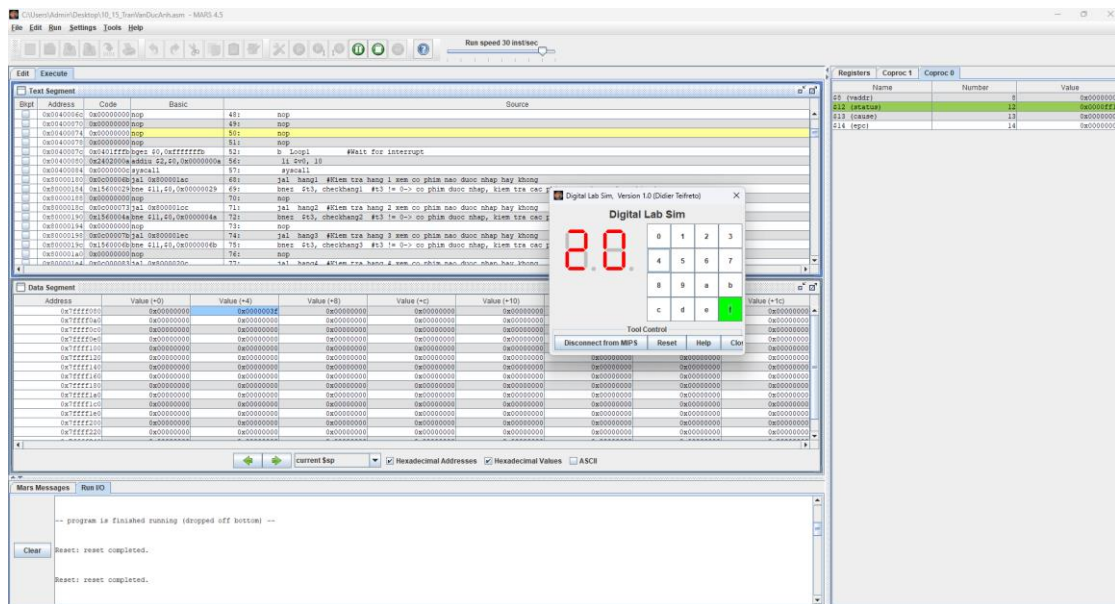


40*2=20

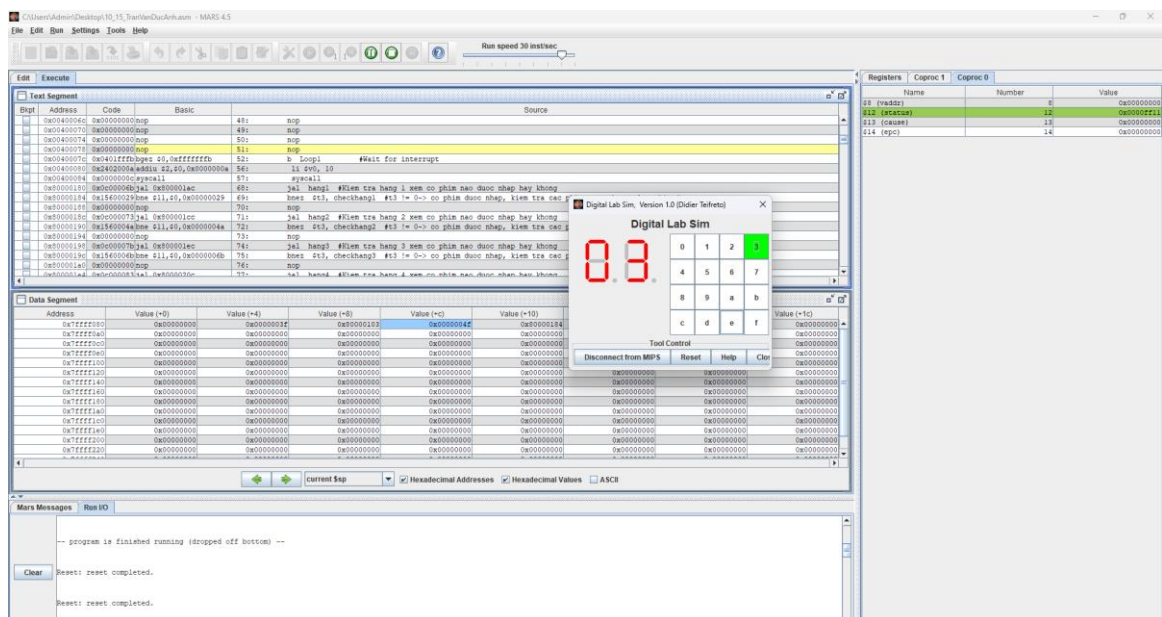
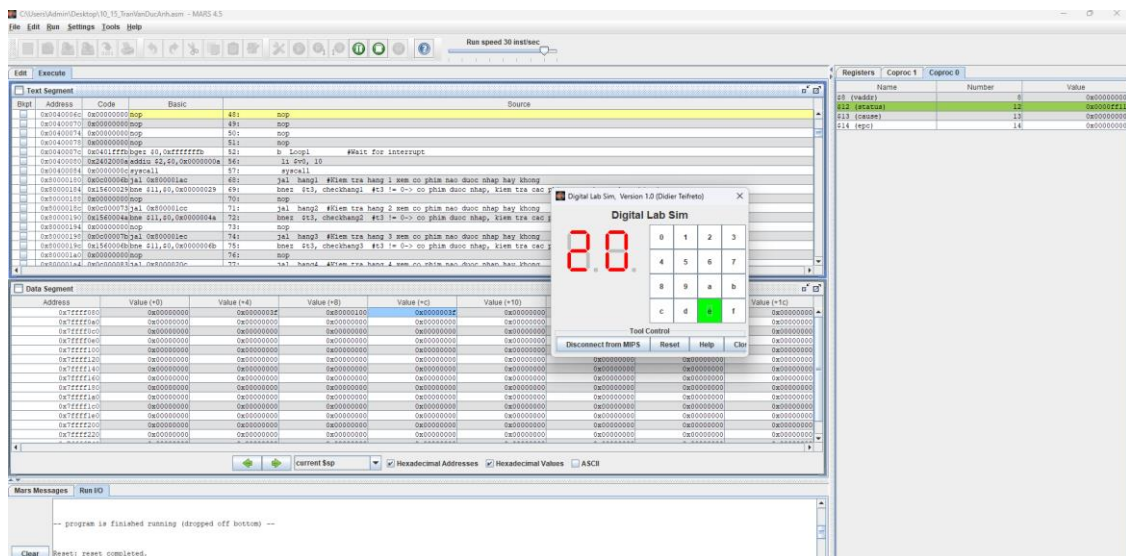


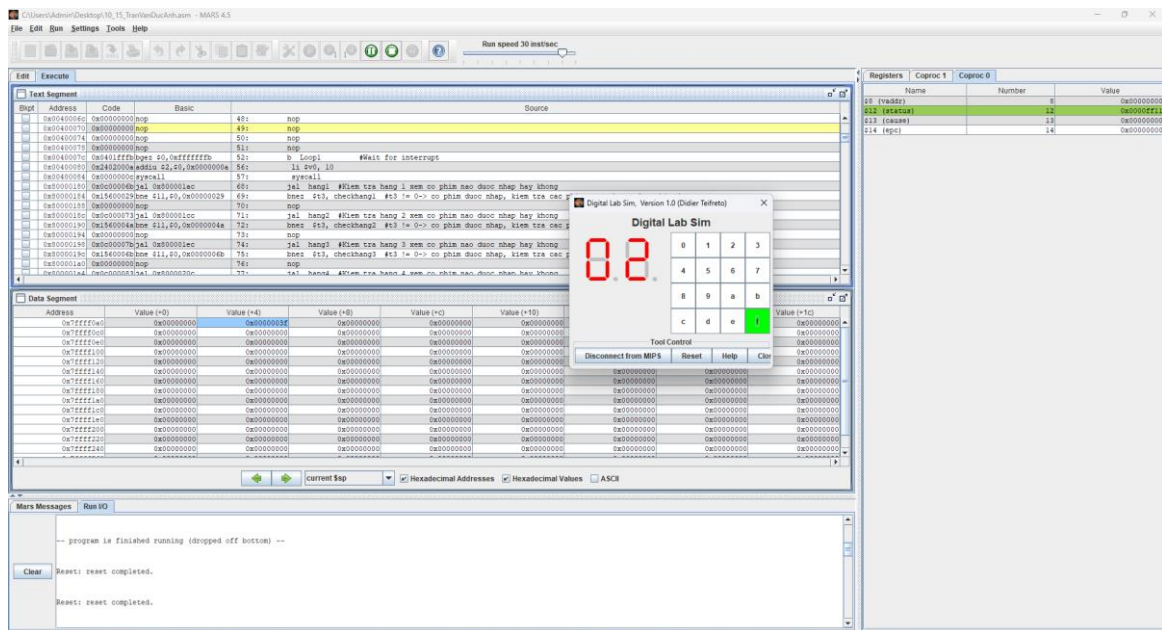
80/4=2





20%3=2





2.5 Code

```
.eqv IN_ADDRESS_HEX_A_KEYBOARD 0xFFFF0012
.eqv OUT_ADDRESS_HEX_A_KEYBOARD 0xFFFF0014
.eqv SEVENSEG_LEFT 0xFFFF0011 # Dia ch icua led 7 doan trai.
.eqv SEVENSEG_RIGHT 0xFFFF0010 # Dia chi cua led 7 doan phai.
```

.data

ma hoa so 0-9 trong digital lab sim
khong: .byte 0x3f

```
mot: .byte 0x6
hai: .byte 0x5b
ba: .byte 0x4f
bon: .byte 0x66
nam: .byte 0x6d
sau: .byte 0x7d
bay: .byte 0x7f
tam: .byte 0x7f
chin: .byte 0x6f
```

```
mess1: .ascii "Ket qua khong ton tai.Khong the chiacho so 0 \n"
.text
```

main:

Khoi_tao_Gia_Tri:

li \$s0,0 #Bien kiem tra xem toan bien nhap vao la so(0) hay toan tu(1)

li \$s1,0 # gia tri bit cua so hien thi o led phai

li \$s2,0 # gia tri bit cua so hien thi o led trai

li \$s3,0 #Bien kiem tra loai toan tu (1: +, 2:-, 3: *,4: / ,5:%)

li \$s4,0 #So thu nhut

```

    li $s5,0 #So thu hai
    li $s6,0 #Ket qua phep tinh: '+','-','*','/','% '
    li $t9, 0 #Bien luu gia tri tam thoi
    li $t0,SEVENSEG_LEFT # khoi tao bien gia tri  cua LED trai
    li $t5,SEVENSEG_RIGHT #khoi tao bien gia tri  cua LED phai
    li $t1, IN_ADDRESS_HEX_KEYBOARD #bien dieukhien hang keyboard vaenable keyboard interrupt
    li $t2, OUT_ADDRESS_HEX_KEYBOARD #bien chuavi tri key nhap vaothe hang va cot
    li $t3, 0x80 # bit dung enable keyboard interrupt va enable kiem tra tung hang keyboard
    sb $t3, 0($t1)
    li $t7, 0 #Gia tri cua so hien tren led
    li $t4, 0 #byte hien thi len led, zero->nine
Gia_Tri_Ban_dau:
    li $t7, 0 #Gia tri cua bit can hien thi ban dau(bat dau voi so 0)
    addi $sp,$sp, 4 #day vao stack
    sb $t7, 0($sp)
    lb $t4, khong #bit dau tien can hien thi
    addi $sp,$sp, 4 #day vao stack
    sb $t4, 0($sp)
Loop1:
    nop
    nop
    nop
    nop
    b Loop1      #Wait for interrupt
end_loop1:

end_main:
li $v0, 10
    syscall

#-----
# Xu ly khi xay ra interupt
# Hien thi so vua bam len den led 7 doan
# Kiem tra tung hang xem co duoc bam hay khong
#-----
.ktext 0x80000180
#-----
# Processing
# Neu hang co phim duoc nhap-> chuyen toi hang do
#-----
jal hang1 #Kiem tra hang 1 xem co phim nao duoc nhap hay khong
    bnez $t3, checkhang1 #t3 != 0-> co phim duoc nhap, kiem tra cac phim trong hang, lay phim do ra
    nop
jal hang2 #Kiem tra hang 2 xem co phim nao duoc nhap hay khong
    bnez $t3, checkhang2 #t3 != 0-> co phim duoc nhap, kiem tra cac phim trong hang, lay phim do ra

```



```

nop
jal hang3 #Kiem tra hang 3 xem co phim nao duoc nhap hay khong
bnez $t3, checkhang3 #t3 != 0-> co phim duoc nhap,kiem tra cac phim trong hang, lay phim do ra
nop
jal hang4 #Kiem tra hang 4 xem co phim nao duoc nhap hay khong
bnez $t3, checkhang4 #t4 != 0-> co phim duoc nhap,kiem tra cac phim trong hang, lay phim do ra
#-----
#Kiem tra xem phim nao duoc nhan trong 4 hang
#-----
hang1:
    addi $sp,$sp,4 # tang dia chi stack
    sw $ra,0($sp) # luu dia chi ra
    li $t3,0x81 #Kich hoat interrupt, bam phim o hang 2
    sb $t3,0($t1)
    lb $t3, 0($t2) #load vi tri phim duoc nhap
    lw $ra,0($sp)
    addi $sp,$sp,-4
    jr $ra
hang2:
    addi $sp, $sp, 4 # tang dia chi stack
    sw $ra, 0($sp) #luu dia chi ra
    li $t3, 0x82 #Kich hoat interrupt, bam phim o hang 2
    sb $t3, 0($t1)
    lb $t3, 0($t2) #load vi tri phim duoc nhap
    lw $ra, 0($sp)
    addi $sp, $sp,-4
    jr $ra
hang3:
    addi $sp, $sp, 4 # tang dia chi stack
    sw $ra, 0($sp) #luu dia chi ra
    li $t3, 0x84 #Kich hoat interrupt, bam phim o hang 3
    sb $t3, 0($t1)
    lb $t3, 0($t2) #load vi tri phim duoc nhap
    lw $ra, 0($sp)
    addi $sp, $sp,-4
    jr $ra
hang4:
    addi $sp, $sp, 4 # tang dia chi stack
    sw $ra, 0($sp) #luu dia chi ra
    li $t3, 0x88 #Kich hoat interrupt, cho phep bam phim o hang 4
    sb $t3, 0($t1)
    lb $t3, 0($t2) #load vi tri phim duoc nhap
    lw $ra, 0($sp)
    addi $sp, $sp,-4

```

```

        jr $ra
#-----
#Convert tu vi tri sang bit
#-----
#-----
# hang 1: co 4 gia tri: 0, 1, 2, 3
#Ma hoa tuong ung la:
# 0: .byte 0x3f
# 1: .byte 0x6
# 2: .byte 0x5b
# 3: .byte 0x4f
#-----
checkhang1:
        beq $t3,0x11, so0 # 0x11->so 0
        beq $t3,0x21, so1 # 0x21->so 1
        beq $t3,0x41, so2 # 0x41->so 2
        beq $t3,0xfffff81, so3 # 0xfffff81-> so 3
so0:
        lb $t4,khong #t4 = so 0(Ma hoa cua '0'tren DigitalLab Sim)
        li $t7,0 #t7= 0
        mul $t9, $t9, 10
        add $t9, $t9, $t7
        j done
so1:
        lb $t4,mot #t4 = so 1(Ma hoa cua '1'tren DigitalLab Sim)
        li $t7,1 #t7 = 1
        mul $t9, $t9, 10
        add $t9, $t9, $t7
        j done
so2:
        lb $t4,hai #t4 = so 2(Ma hoa cua '2'tren DigitalLab Sim)
        li $t7,2 #t7 = 2
        mul $t9, $t9, 10
        add $t9, $t9, $t7
        j done
so3:
        lb $t4,ba #t4 = so 3 (Ma hoa cua '3'tren DigitalLab Sim)
        li $t7,3 #t7 = 3
        mul $t9, $t9, 10
        add $t9, $t9, $t7
        j done
#-----
# hang 2: co 4 gia tri: 4, 5, 6, 7
#Ma hoa tuong ung la:

```

```

# 4: .byte 0x66
# 5: .byte 0x6d
# 6: .byte 0x7d
# 7: .byte 0x7
#-----
checkhang2:
    beq $t3,0x12, so4 # 0x12->so 4
    beq $t3,0x22, so5 # 0x22->so 5
    beq $t3,0x42, so6 # 0x42->so 6
    beq $t3,0xfffff82, so7 # 0xfffff82-> so 7
so4:
    lb $t4, bon #t4 = so 4(Ma hoa cua '4'tren DigitalLab Sim)
    li $t7, 4 #t7= 4
    mul $t9, $t9, 10
    add $t9, $t9, $t7
    j done
so5:
    lb $t4, nam #t4 = so 5(Ma hoa cua '5'tren DigitalLab Sim)
    li $t7, 5 #t7 = 5
    mul $t9, $t9, 10
    add $t9, $t9, $t7
    j done
so6:
    lb $t4, sau #t4 = so 6(Ma hoa cua '6'tren DigitalLab Sim)
    li $t7,6 #t7 = 6
    mul $t9, $t9, 10
    add $t9, $t9, $t7
    j done
so7:
    lb $t4, bay #t4 = so 7 (Ma hoa cua '7'tren DigitalLab Sim)
    li $t7, 7 #t7 = 7
    mul $t9, $t9, 10
    add $t9, $t9, $t7
    j done
#-----
#hang3: co 4 gia tri:8, 9, a, b
#Ma hoa tuong ung la:
# eight: .byte 0x7f
# nine: .byte 0x6f
#-----
checkhang3:
    beq $t3,0x14, so8 # 0x12->so 8
    beq $t3,0x24, so9 # 0x22->so 9
    beq $t3,0x44, so_a # 0x42->phim a

```



```

        beq $t3,0xffffffff84, so_b # 0xffffffff82-> phim b
so8:
        lb $t4,tam #t4 = eight (Ma hoa cua '8'tren DigitalLab Sim)
        li $t7,8 #t7= 8
        mul $t9, $t9, 10
        add $t9, $t9, $t7
        j done
so9:
        lb $t4,chin #t4 = nine (Ma hoa cua '9'tren DigitalLab Sim)
        li $t7,9 #t7 = 9
        mul $t9, $t9, 10
        add $t9, $t9, $t7
        j done
#-----
#Truong hop phim a (phep cong)
#-----
so_a:
        addi $s0,$s0, 1 #s0 = 1-> toan tu duoc nhap vao
        bne $s3,0, set_next_operator
        addi $s3,$zero, 1 #s3 = 1-> phep cong
        j luugiatriSodautien #chuyen denham chuyen 2 byte dang hien tren 2 ledthanh so de tinhtoan
#-----
#Truong hop phim b (phep tru)
#-----
so_b:
        addi $s0,$s0, 1 #s0 = 1-> toan tu duoc nhap vao
        bne $s3,0, set_next_operator
        addi $s3,$zero, 2 #s3 = 2-> phep tru
        j luugiatriSodautien
#-----
#hang4: co 4 gia tri:c, d, e, f
#-----
checkhang4:
        beq $t3,0x18, so_c # 0x18->phim c
        beq $t3,0x28,so_d # 0x28->phim d
        beq $t3,0x48,so_e # 0x48-> phim e
        beq $t3,0xffffffff88, so_f # 0xffffffff88-> phim f
#-----
#Truong hop phim c (phep nhan)
#-----
so_c:
        addi $s0,$s0, 1 #s0 = 1-> toan tu duoc nhap vao
        bne $s3,0, set_next_operator
        addi $s3,$zero, 3 #s3 = 3-> phep nhan

```

```

j luugiatriSodautien #chuyen den ham chuyen 2 byte dang hien tren 2 led thanh so de tinh toan
#-----
#Truong hop phim d (phep chia)
#-----
so_d:
    addi $s0, $s0, 1 #s0 = 1-> toan tu duoc nhap vao
    bne $s3, 0, set_next_operator
    addi $s3, $zero, 4 #s3 = 4-> phep chia lay phan nguyen
    j luugiatriSodautien
#-----
#Truong hop phim d (phep chia du)
#-----
so_e:
    addi $s0, $s0, 1 #s0 = 1-> toan tu duoc nhap vao
    bne $s3, 0, set_next_operator
    addi $s3, $zero, 5 #s3 = 4-> phep chia lay phan nguyen
    j luugiatriSodautien
#-----
#Truong hop phim f (=)
#-----
so_f:
    addi $s5, $t9, 0
    j luugiatriSothuhaiVatinhtoan
#-----
#Tinh so dau hien thi tren den led
#-----
luugiatriSodautien:
    addi $s4, $t9, 0
    li $t9, 0
    j done
#-----
#Tinh so thu hai hien thi tren den led trong 2 so
#-----
luugiatriSothuhaiVatinhtoan:
    beq $s3, 1, cong # s3 = 1-> cong
    beq $s3, 2, tru # s3 = 2-> tru
    beq $s3, 3, nhan # s3 = 3-> nhan
    beq $s3, 4, chia # s3 = 4-> chia
    beq $s3, 5, chia_du # s3 = 5-> chia du
cong:
    add $s6, $s5, $s4
    li $s3, 0
    li $t9, 0
li $s7, 100

```

```

        div $s6, $s7
        mfhi $s6 #Lay 2 gia tri cuoi cua ket qua
        j hienketquatrenLed #Hien thi ket qua tren led
        nop
tru:
sub $s6, $s4, $s5
    li $s3, 0
    li $t9, 0
li $s7, 100
div $s6, $s7
mfhi $s6 #Lay 2 giatri cuoi cua ket qua
j hienketquatrenLed #Hien thi ket qua tren led
nop
nhan:
mul $s6, $s4, $s5
li $s3, 0
li $t9, 0
li $s7, 100
    div $s6, $s7
    mfhi $s6 #Lay 2 chuso sau cung cuaket qua in ra
    j hienketquatrenLed #Hien thiket qua tren led
    nop
chia:
    beq $s5, 0, chia_0 # kiem traxem so chia co phai bang 0 hay khong
li $s3, 0
div $s4, $s5
mflo $s6
li $t9, 0
li $s7, 100
div $s6, $s7
mfhi $s6 #Lay 2 gia tri cuoi cua ket qua
j hienketquatrenLed #Hien thi ket qua tren led
    nop
chia_du:
    beq $s5, 0, chia_0 # kiem traxem so chia co phai bang 0 hay khong
li $s3, 0
div $s4, $s5
mfhi $s6
li $t9, 0
li $s7, 100
div $s6, $s7
mfhi $s6 #Lay 2 gia tri cuoi cua ket qua
j hienketquatrenLed #Hien thi ket qua tren led
    nop

```

```

#-----
# Neu $s5 = 0 in ra: "Khong the chia cho so 0 \n"
#-----
chia_0:
    li $v0, 5
    la $a0, mess1
    li $a1, 0
    syscall
    j reset_led

#-----
#Hien thi ket qua tren den led
#So 'ab'
# Den trai = a = ab div 10
# Den phai = b = ab mod 10
#-----
hienketquatrenLed:
    li $t8,10 # Gia tritung gian = 10
    div $s6,$t8 # $s6 =a
    mflo $t7 # $t7 =result
    jal check #chuyen denham chuyen t7 thanhbit hien thi len led
    sb $t4,0($t0) # hien thilen led trai
    addi $sp, $sp, 4
    sb $t7,0($sp) #day gia tri bit nay vao stack
    addi $sp,$sp, 4
    sb $t4,0($sp) #day bit nay vao stack
    add $s2,$t7, $zero #s1 = gia tri bit led phai
    mfhi $t7 #t7 = remainder
    jal check #convert t7thanh bit hien thilen led
    sb $t4,0($t5) #hien thi len led phai
    add $sp, $sp, 4
    sb $t7,0($sp) # day giatri bit nay vaostack
    add $sp,$sp, 4
    sb $t4,0($sp) # day bitnay vao stack
    add $s1,$t7, $zero # s1 = gia tri bit led phai
    j reset_led # ham resetlai led
check:
    addi $sp,$sp, 4
    sw $ra,0($sp)
    beq $t7,0, check_0 # t7 =0-> Hien thi so0 tren thanh led
    beq $t7,1, check_1 # t7 = 1-> Hien thi so 1tren thanh led
    beq $t7,2, check_2 # t7 =2-> Hien thi so2 tren thanh led
    beq $t7,3, check_3 # t7 =3-> Hien thi so3 tren thanh led
    beq $t7,4, check_4 # t7 =4-> Hien thi so4 tren thanh led
    beq $t7,5, check_5 # t7 =5-> Hien thi so5 tren thanh led

```

```

beq $t7,6, check_6 # t7 =6-> Hien thi so6 tren thanh led
beq $t7,7, check_7 # t7 =7-> Hien thi so7 tren thanh led
beq $t7,8, check_8 # t7 =8-> Hien thi so8 tren thanh led
beq $t7,9, check_9 # t7 =9-> Hien thi so9 tren thanh led

```

```

#-----

```

```

#Chuyen giatri thanh bit hientren thanh led

```

```

#-----

```

```

check_0:

```

```

    lb $t4, khong

```

```

    j finish_check

```

```

check_1:

```

```

    lb $t4, mot

```

```

    j finish_check

```

```

check_2:

```

```

    lb $t4, hai

```

```

    j finish_check

```

```

check_3:

```

```

    lb $t4, ba

```

```

    j finish_check

```

```

check_4:

```

```

    lb $t4, bon

```

```

    j finish_check

```

```

check_5:

```

```

    lb $t4, nam

```

```

    j finish_check

```

```

check_6:

```

```

    lb $t4, sau

```

```

    j finish_check

```

```

check_7:

```

```

    lb $t4, bay

```

```

    j finish_check

```

```

check_8:

```

```

    lb $t4, tam

```

```

    j finish_check

```

```

check_9:

```

```

    lb $t4, chin

```

```

    j finish_check

```

```

finish_check:

```

```

    lw $ra, 0($sp)

```

```

    addi $sp, $sp,-4

```

```

    jr $ra

```

```

#-----

```

```

# Hoan thanh xong 1 so-> reset_led

```

```

#-----
done:
    beq $s0,1,reset_led # s0 = 1-> toan tu-> chuyen den ham reset led
    nop
#-----
#ham hien thi bit len led ben trai
#-----
Hienledtrai:
    lb $t6, 0($sp) #load bit hien thi led tu stack
    add $sp, $sp,-4
    lb $t8, 0($sp) #load gia tri cua bit nay
    add $sp, $sp,-4
    add $s2, $t8, $zero #s2 = gia tri bit led trai
    sb $t6, 0($t0) # hien thi len led trai
#-----
#ham hien thi bit len led ben phai
#-----
Hienledphai:
    sb $t4, 0($t5) # hien thi bit len led phai
    add $sp, $sp,4
    sb $t7, 0($sp) #day gia tri bit nay vao stack
    add $sp, $sp,4
    sb $t4, 0($sp) #day bit nay vao stack
    add $s1, $t7, $zero #s1 = gia tri bit led phai
    j kethuc
reset_led:
    li $s0, 0 #s0 = 0-> doi nhap so tiep theo trong 2 so
    li $t8, 0
    addi $sp, $sp, 4
    sb $t8, 0($sp)
    lb $t6, khong
    addi $sp, $sp, 4
    sb $t6, 0($sp)
kethuc:
    j quayve
    nop
# day bit zero vao stack
#-----
#-----
quayve: add $t9,$t9,$s6
li $s6,0
    la $a3, Loop1
    mtc0 $a3, $1
    eret

```

set_next_operator:

