**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

  



**BÁO CÁO CUỐI KỲ**

**MÔN THỰC HÀNH KIẾN TRÚC MÁY TÍNH KỲ 20232**

**Giảng viên hướng dẫn: TS. Lê Bá Vui Sinh viên:**

**Nguyễn Đức Tuân - 20184217**

**Nguyễn Thành Vinh – 20225779**

*Hà Nội, tháng ... ngày ... năm 2024*

## MỤC LỤC

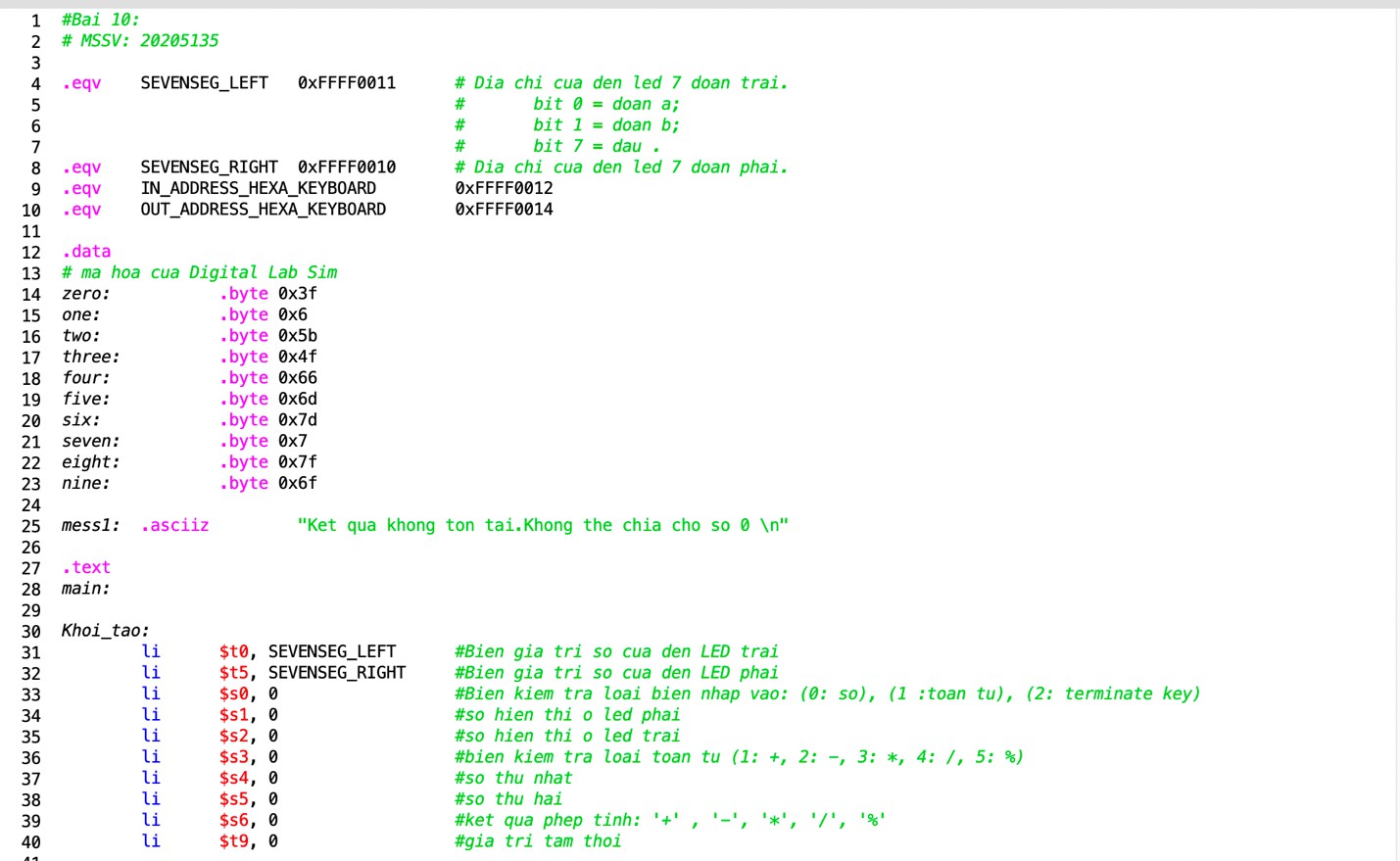
1. **Phân chia công việc**
2. **Project 10: Máy tính bỏ túi**
3. **Project 3: Kiểm tra tốc độ và độ chính xác khi gõ văn bản**

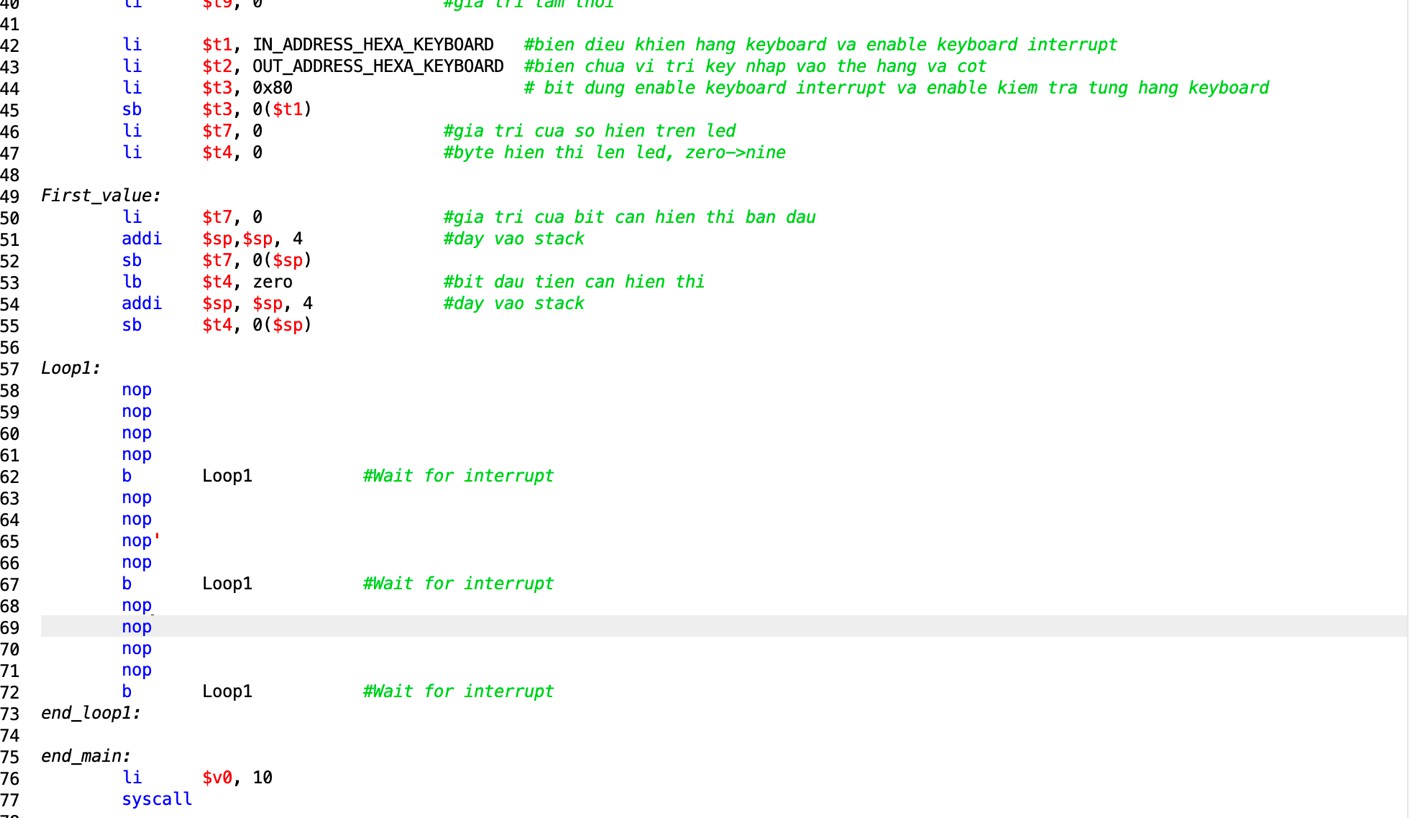
# Phân chia công việc

* + Nguyễn Đức Tuân - 20184217: Project 10
  + Nguyễn Thành Vinh - 20225779: Project 3
  + Công cụ sử dụng : Mars4\_5

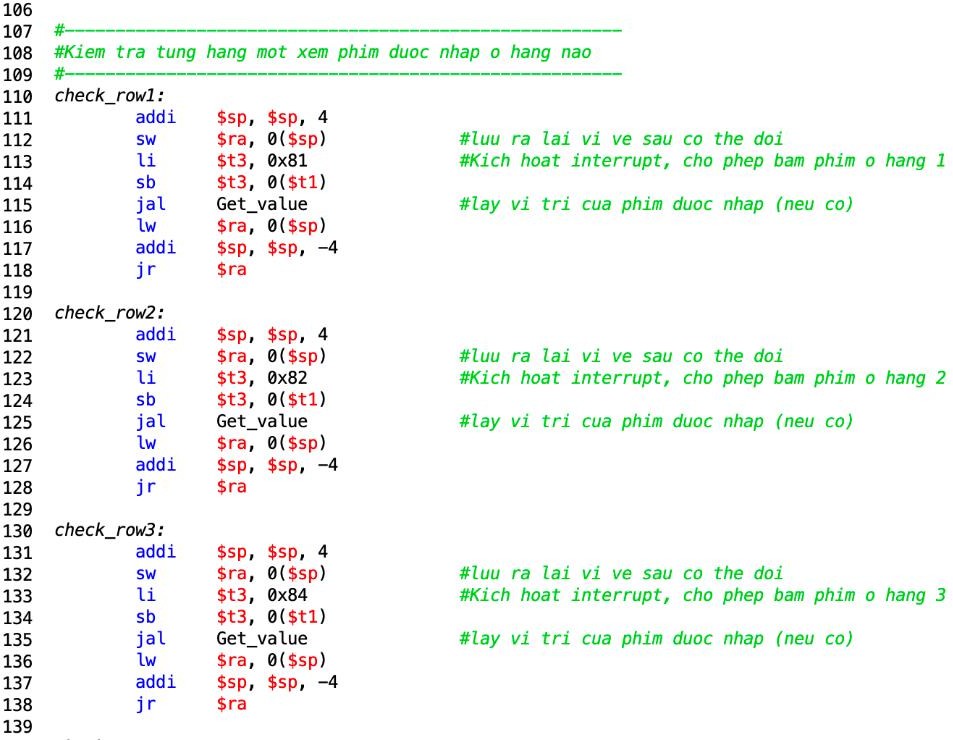
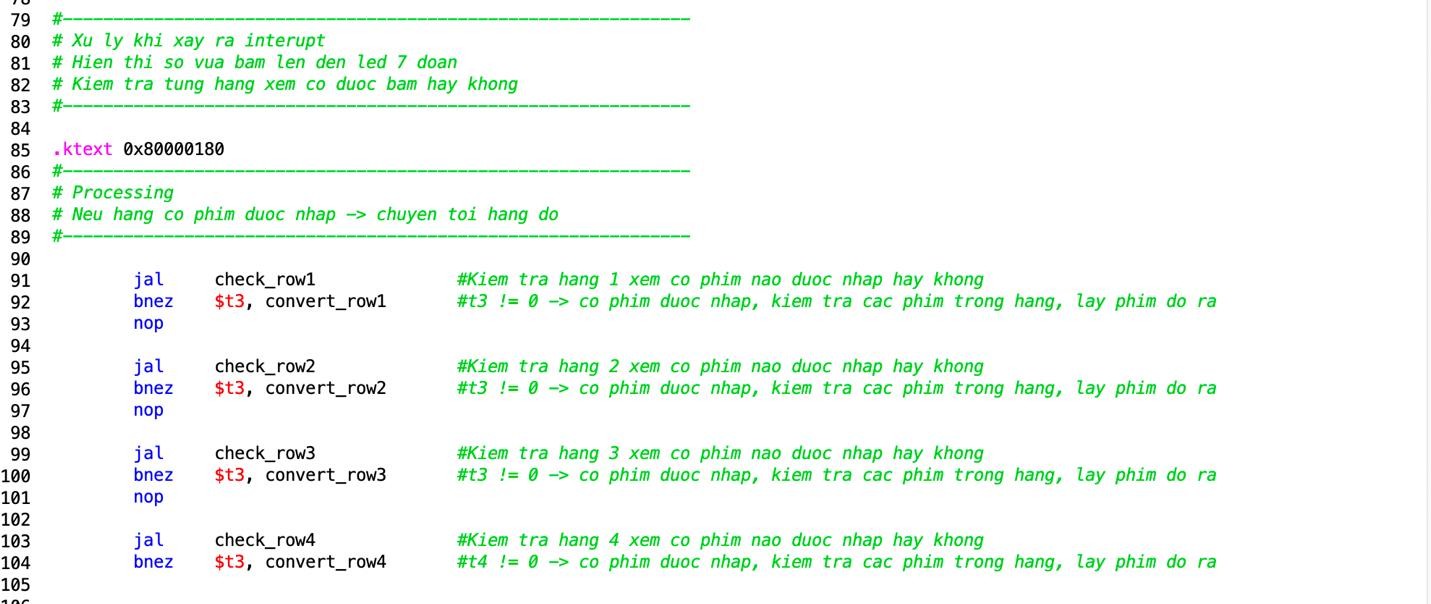
# 2. Project 10: Máy tính bỏ túi

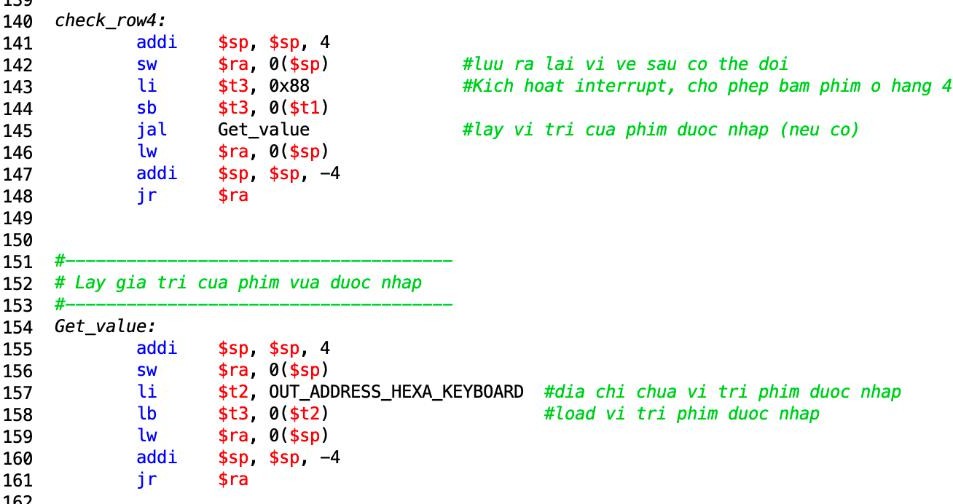
* 1. **Hướng dẫn chạy chương trình**
     + Mở công cụ Digital Lab Sim, Connect to MIPS và chạy chương trình.
     + Nhập các số bằng cách nhập từ các phím trên công cụ Digital Lab Sim và màn hình led sẽ hiển thị 2 chữ số cuối.
     + Nhập xong số đầu tiên thì nhập toán tử cộng, trừ, nhân, chia tương ứng với a, b, c, d, e hiện trên phím của công cụ Digital Lab Sim.
     + Nhập số thứ hai.
     + Nhấn f (tương đương dấu ‘=’) để hiển thị 2 chữ số cuối cùng của kết quả.
  2. **Phân tích cách thức thực hiện**
     + Nhập từ bàn phím ảo của công cụ Digital Lab Sim và xác định mã của phím vừa nhận.
     + Thực hiện kiểm tra:
       - 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.
       - 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) và chuyển sang nhập số thứ hai.
     + Nếu toán tử nhập vào là ‘=’ (dấu bằng) -> hiển thị kết quả ra màn hình trên thanh led 7 đoạn của Digital Lab Sim.
     + Những lưu ý:
       - 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.
       - 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.
       - 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).
       - 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).
       - 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.
  3. **Code**
* Khởi tạo

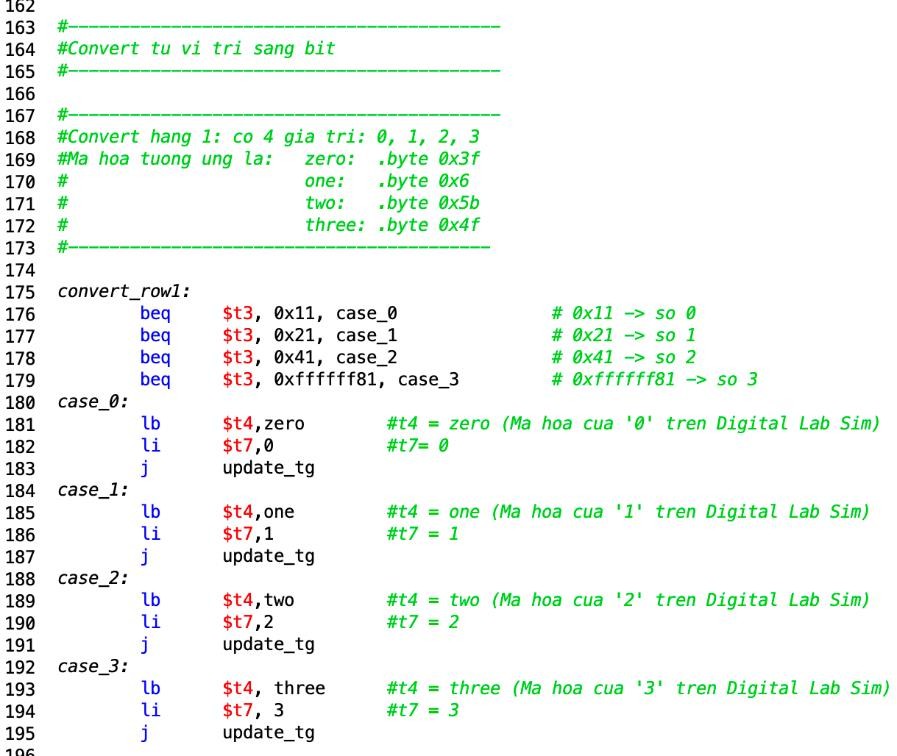


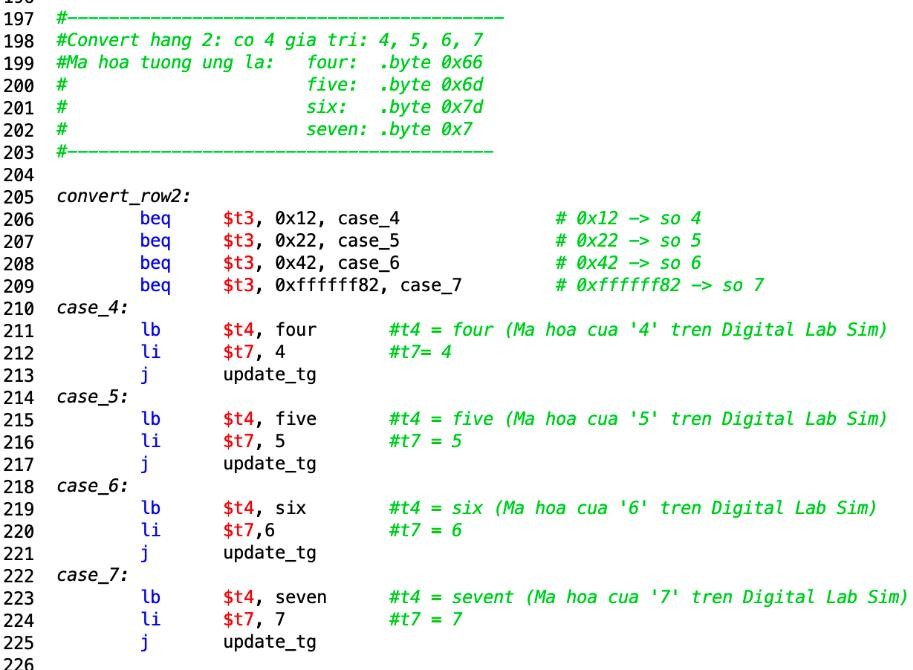


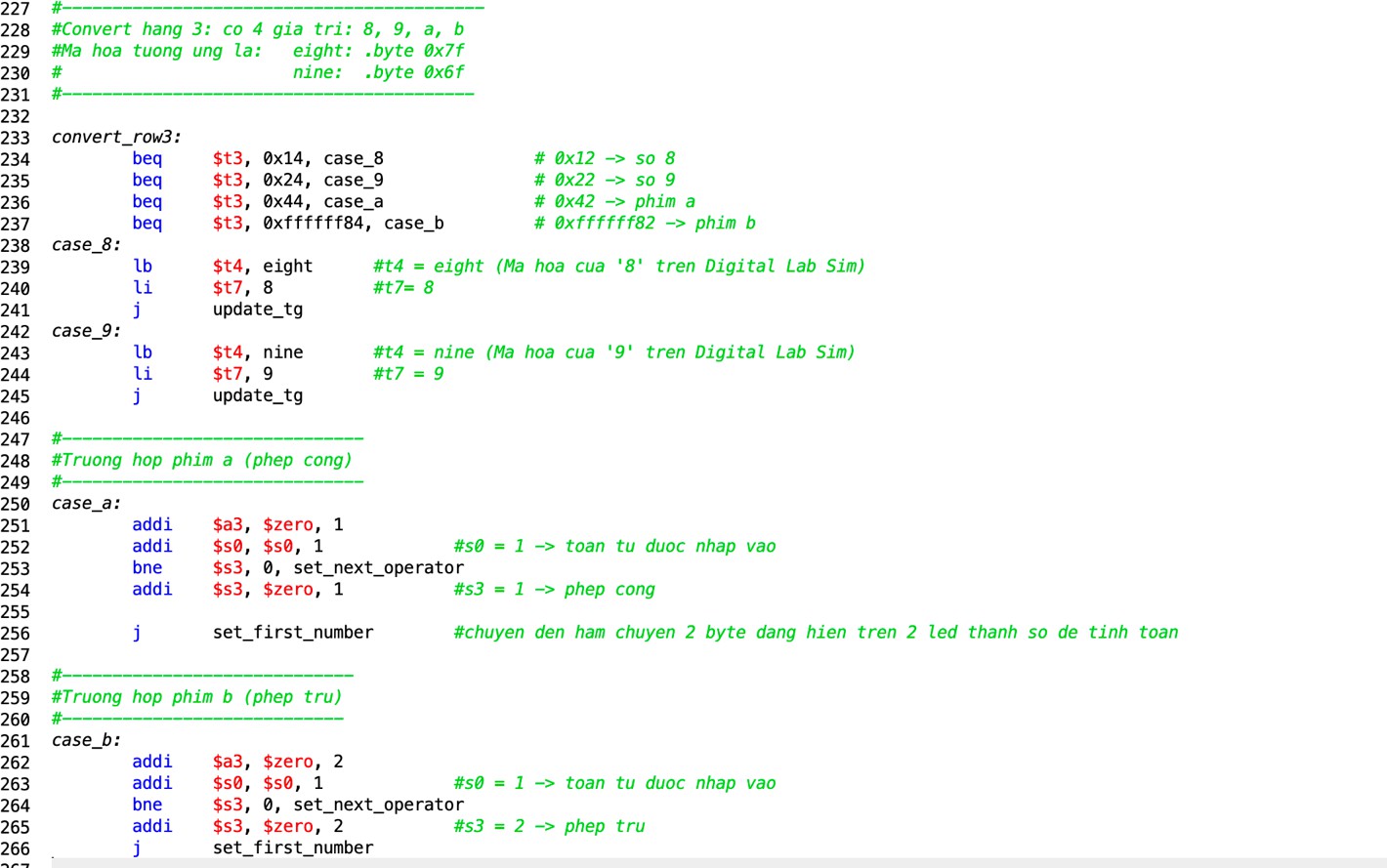
* Kiểm tra xem số vừa được nhập vào

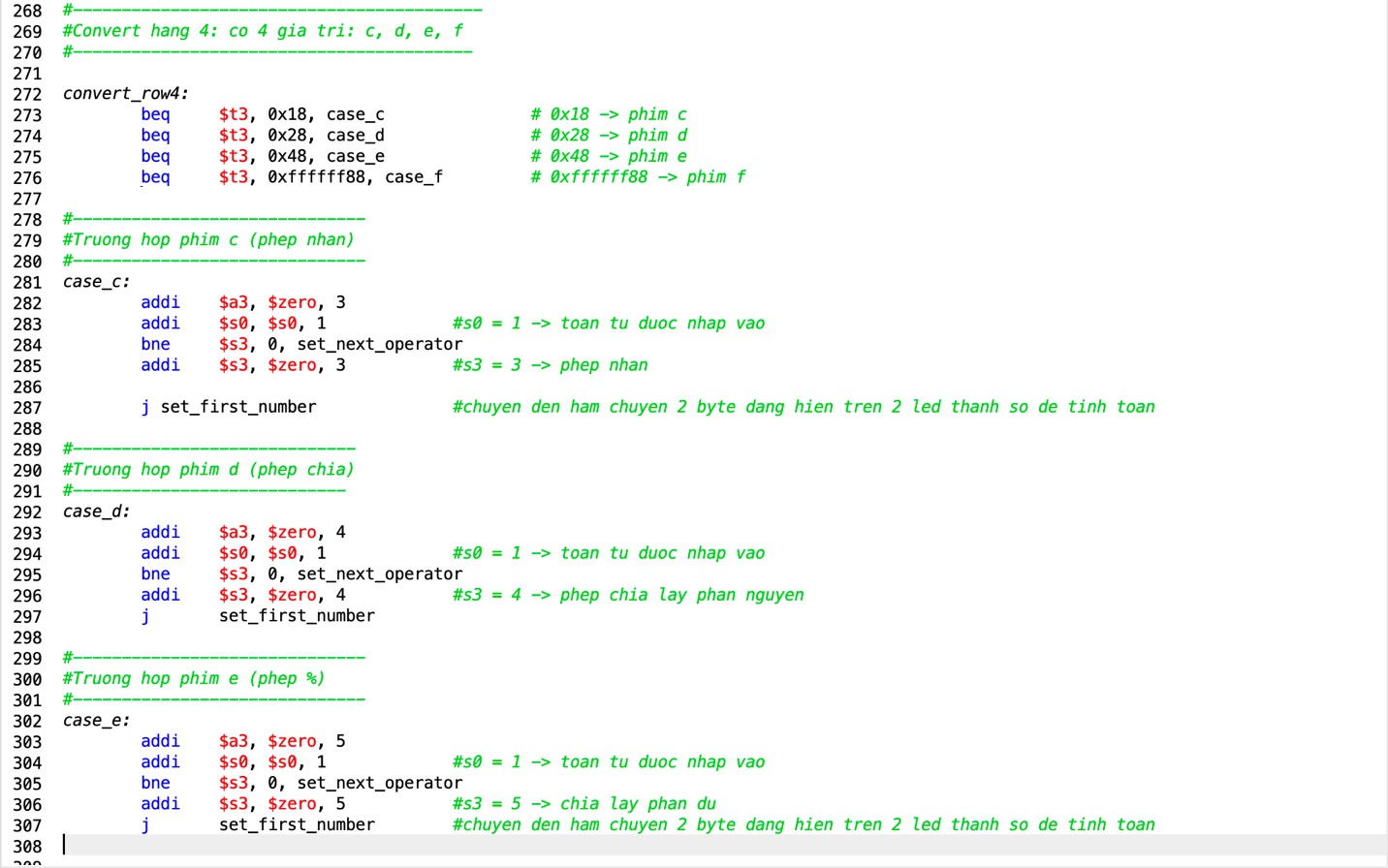




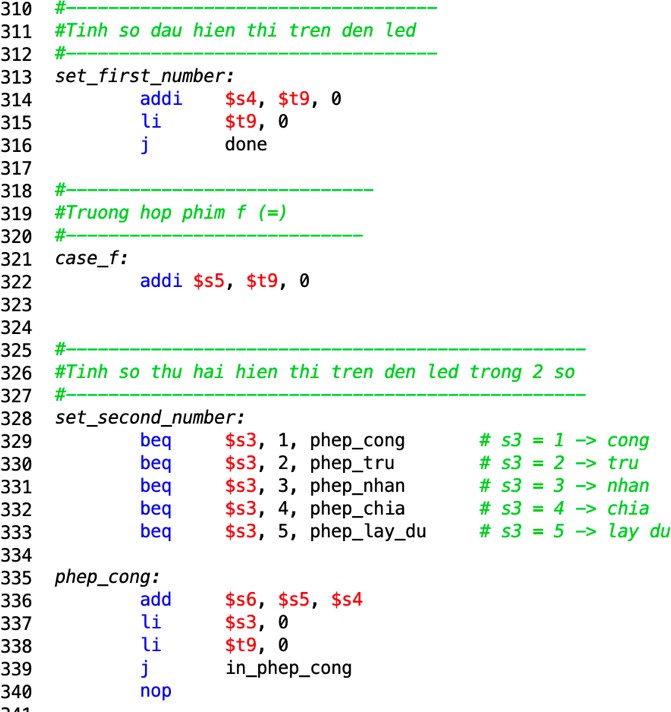


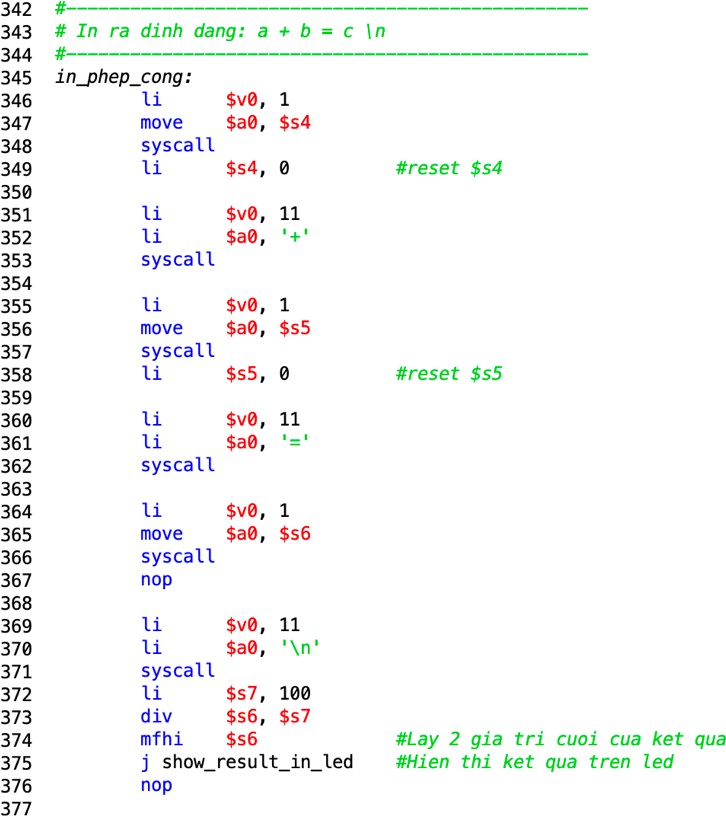


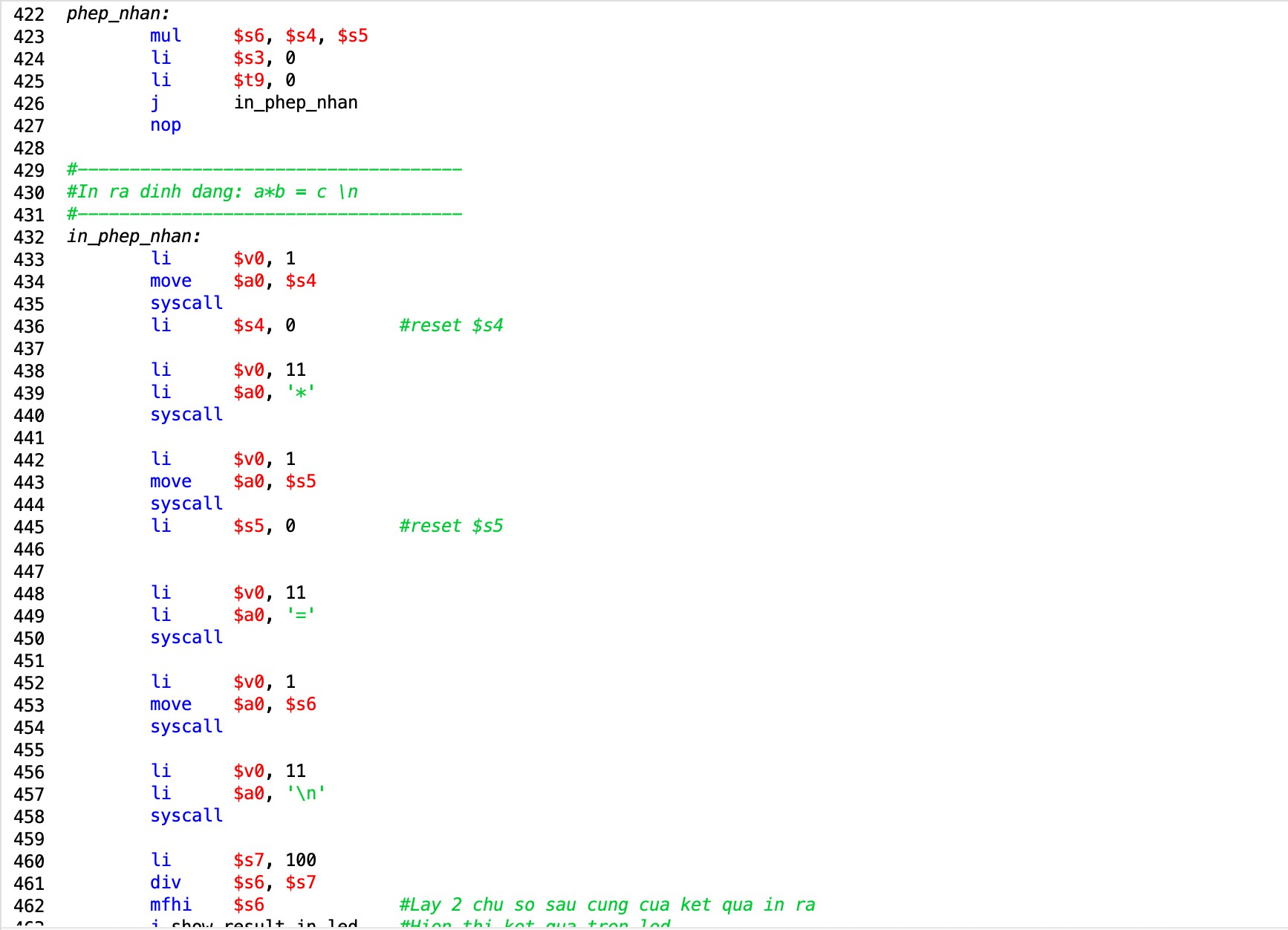


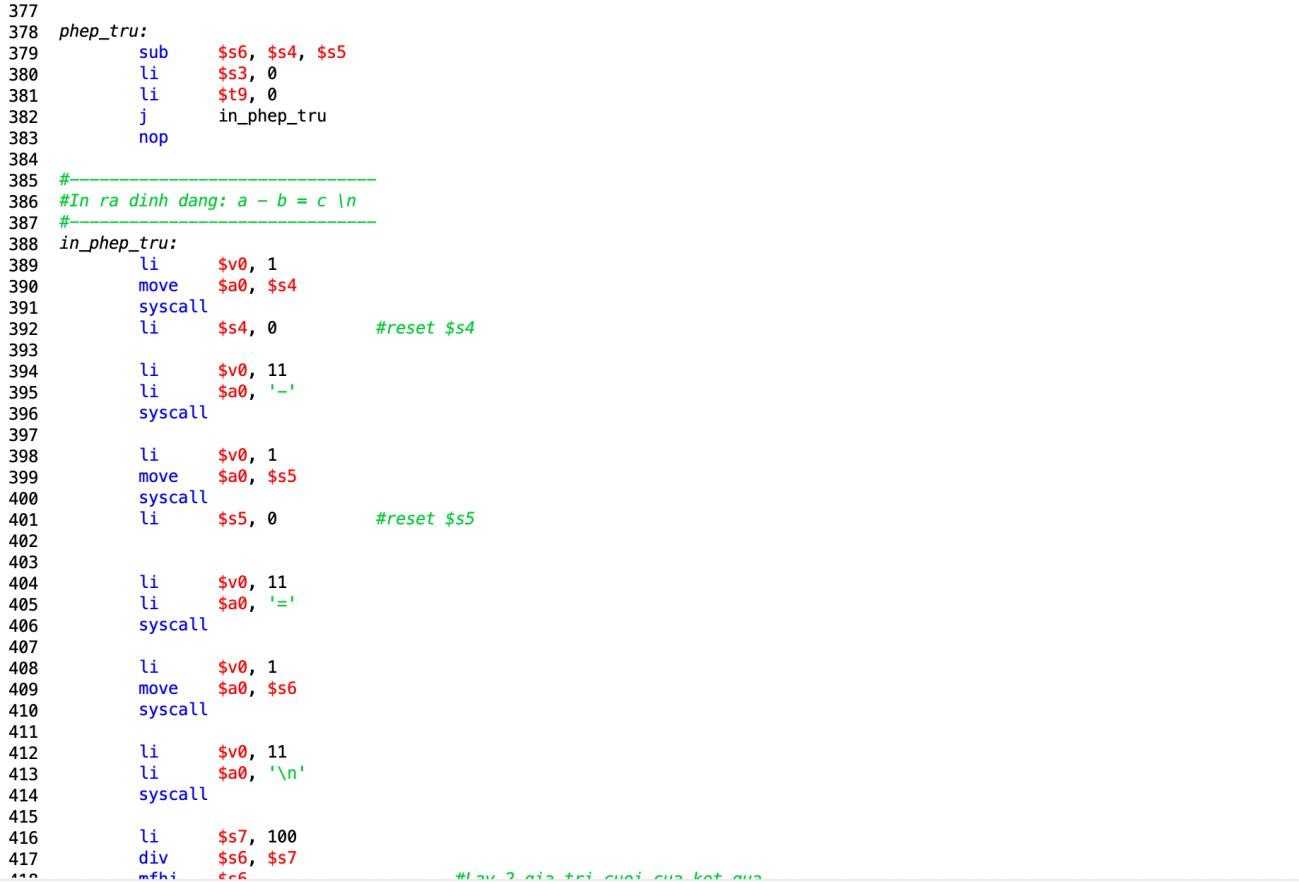


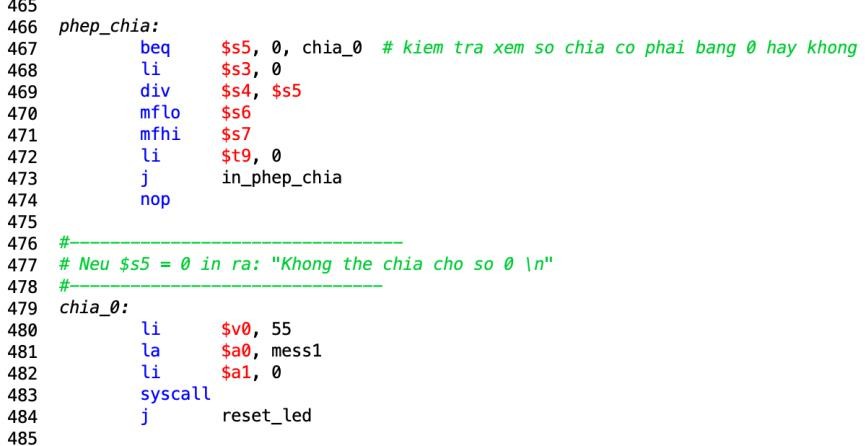
* Thực hiện phép tính.

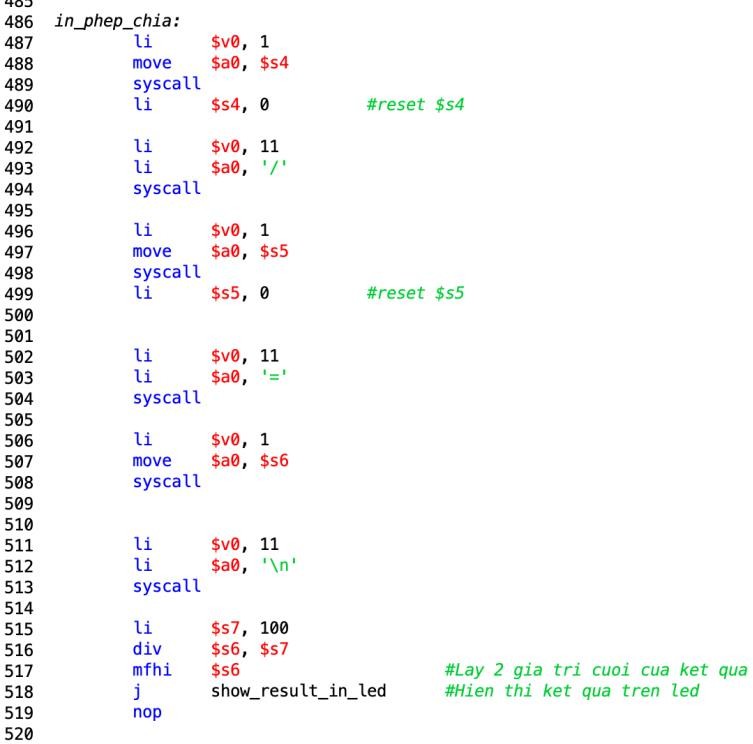


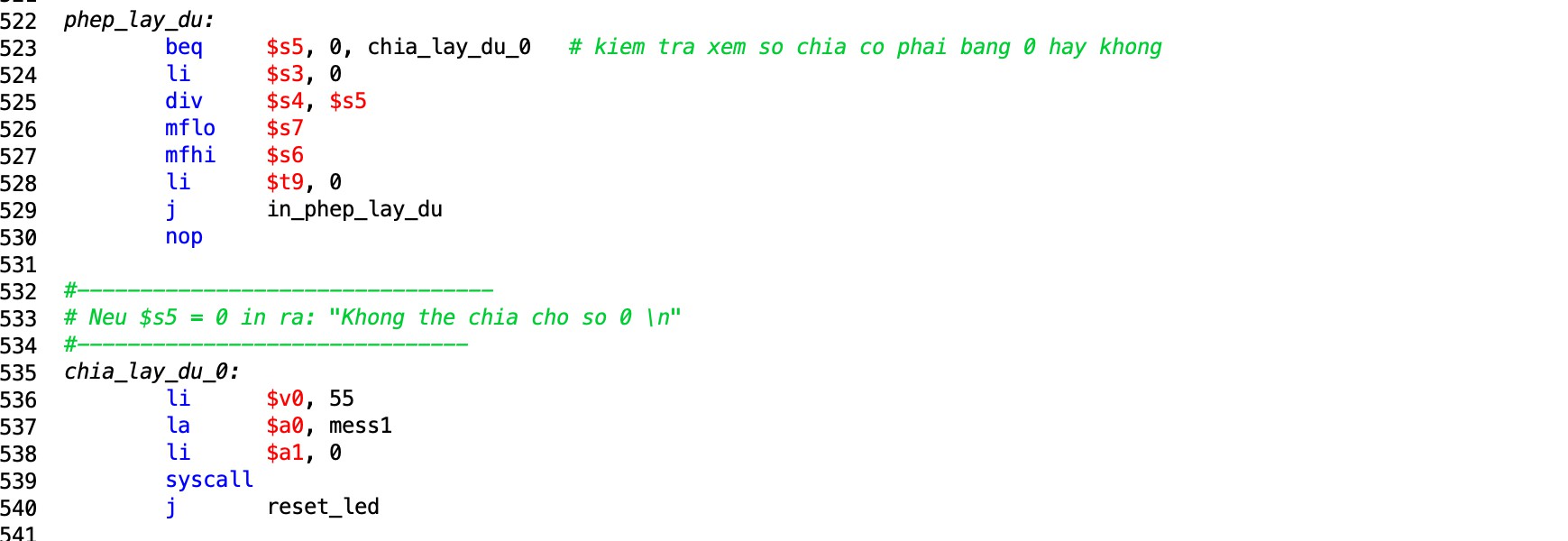


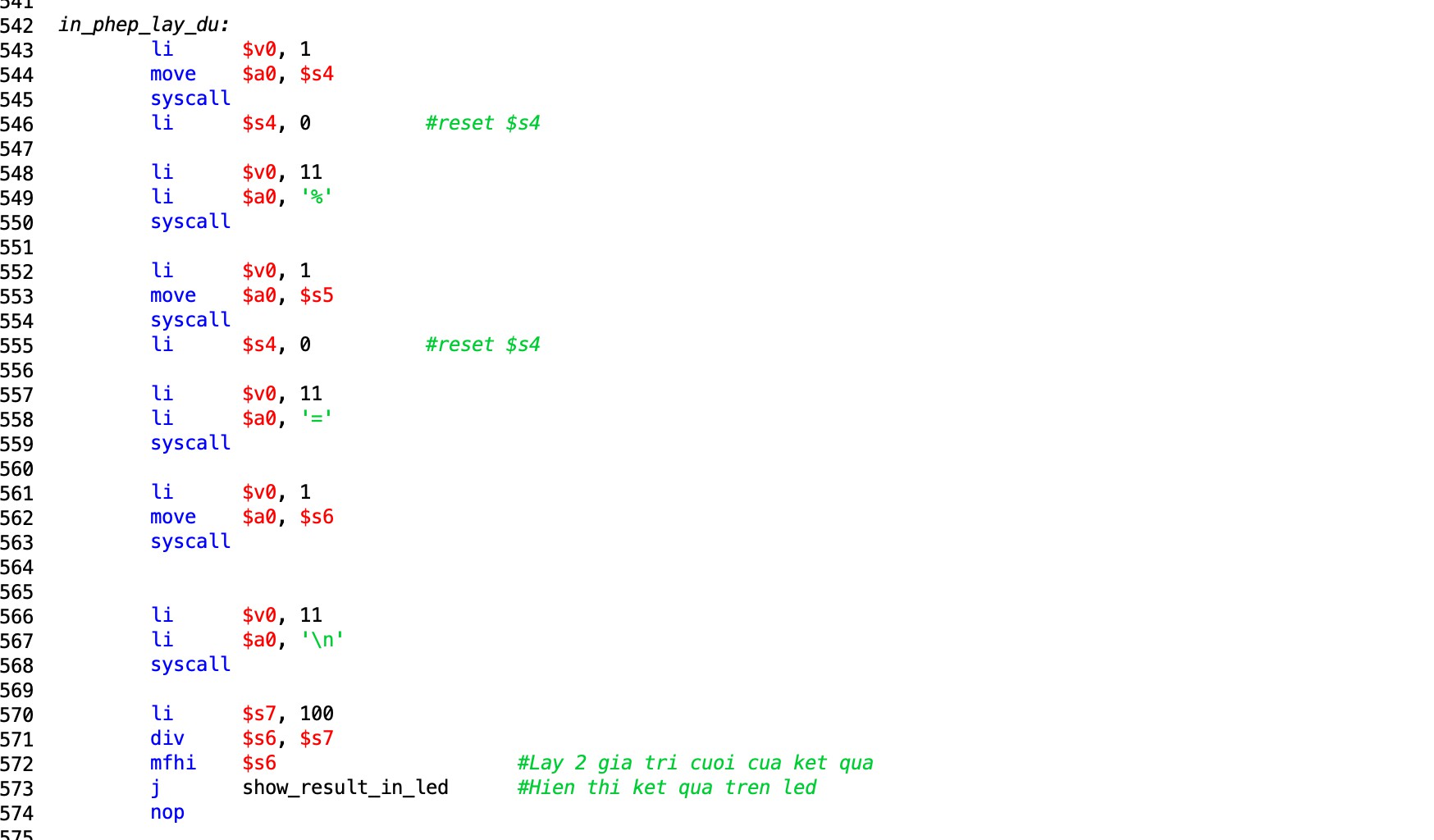


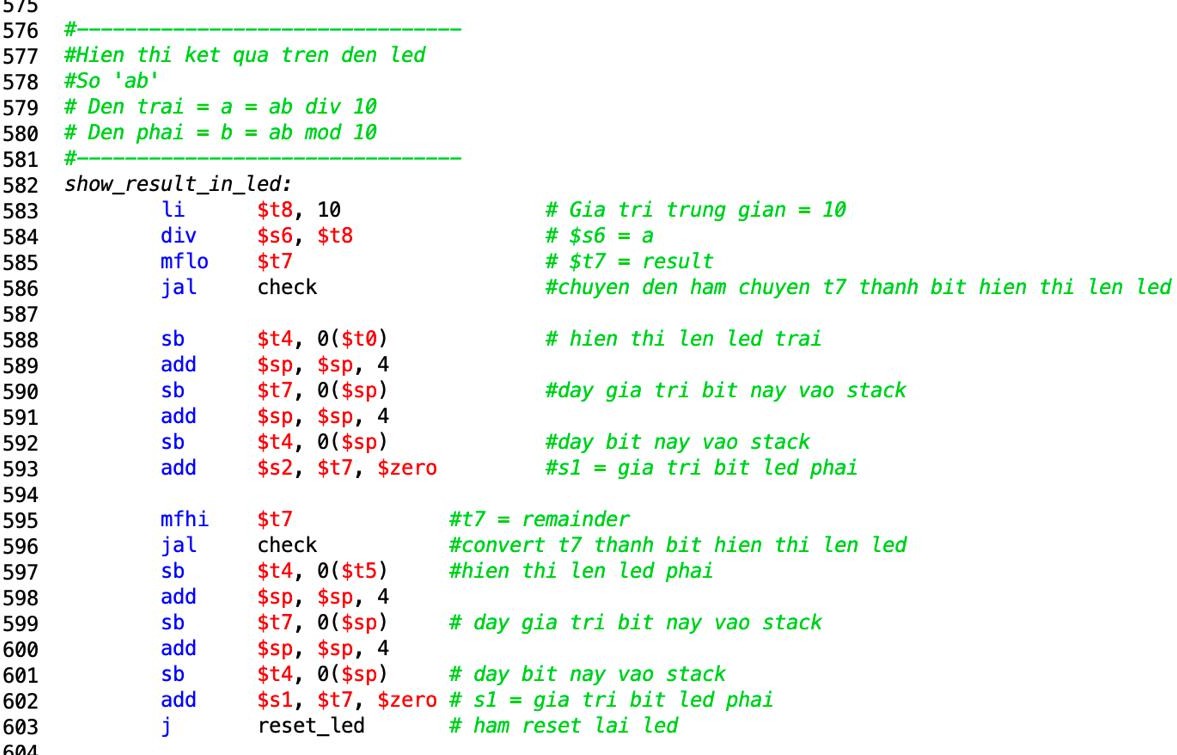


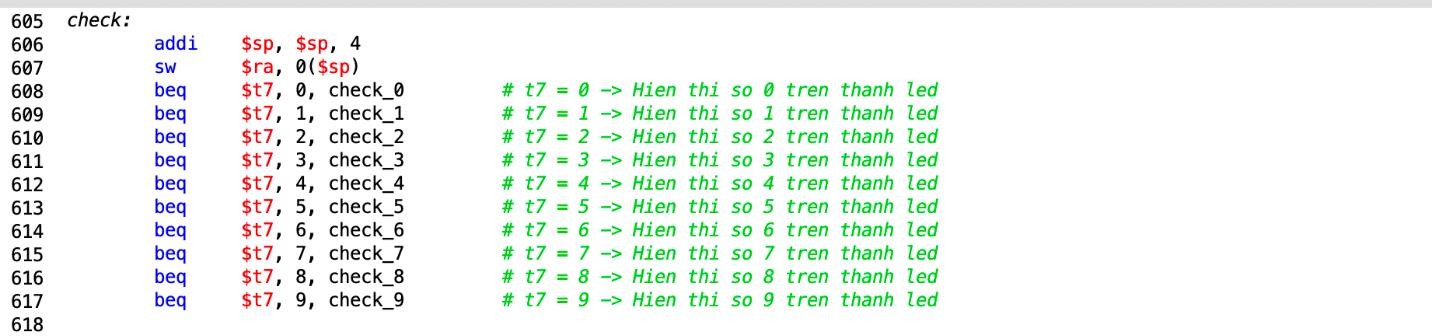


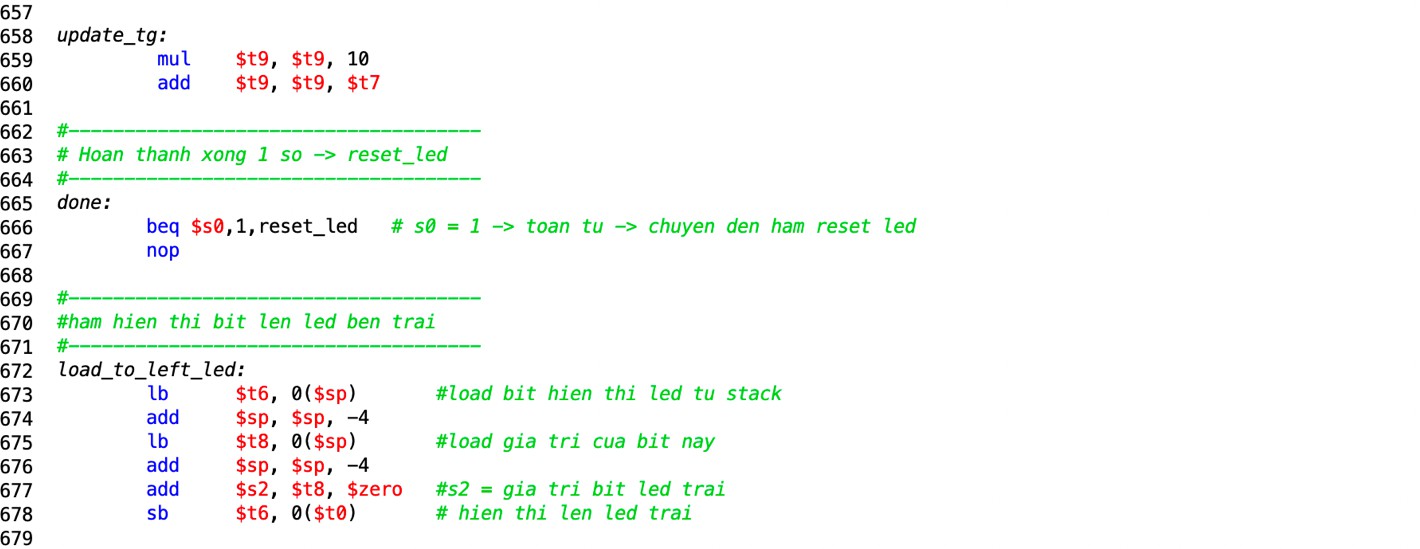


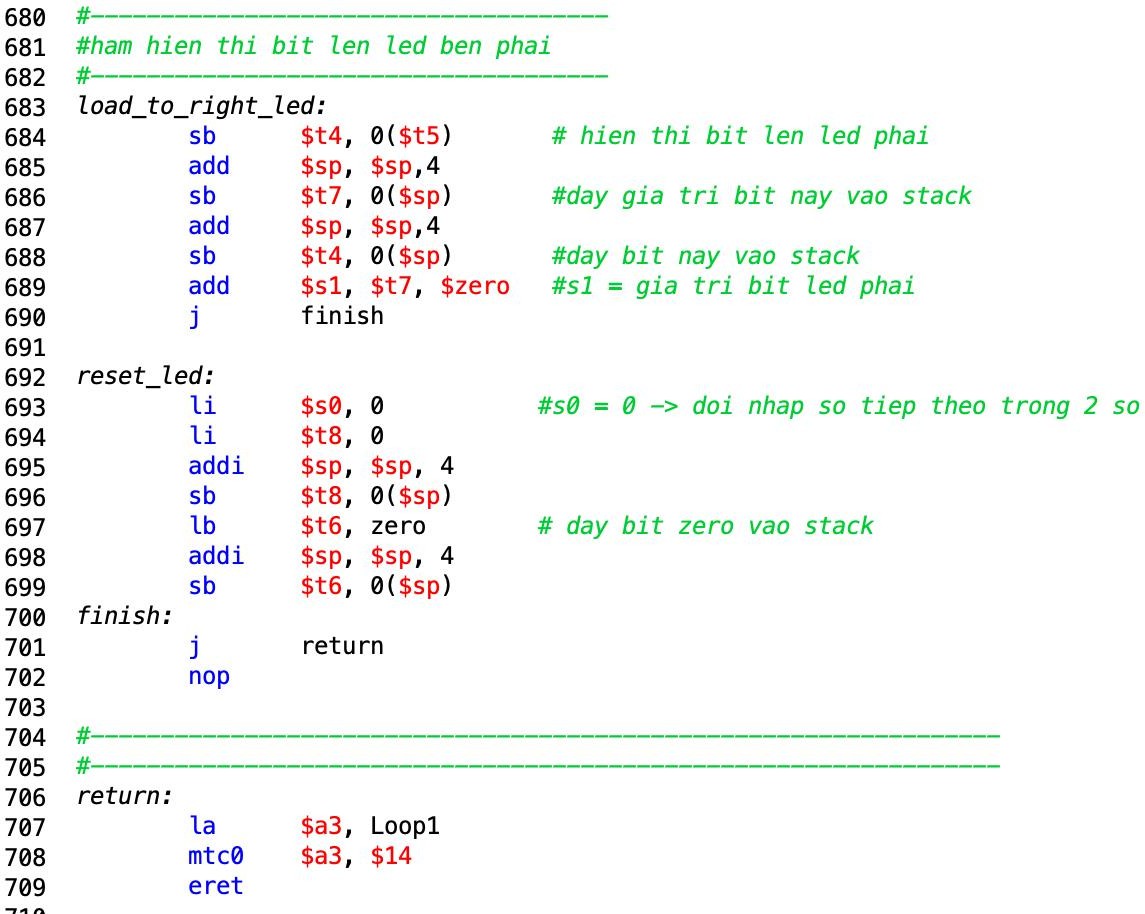


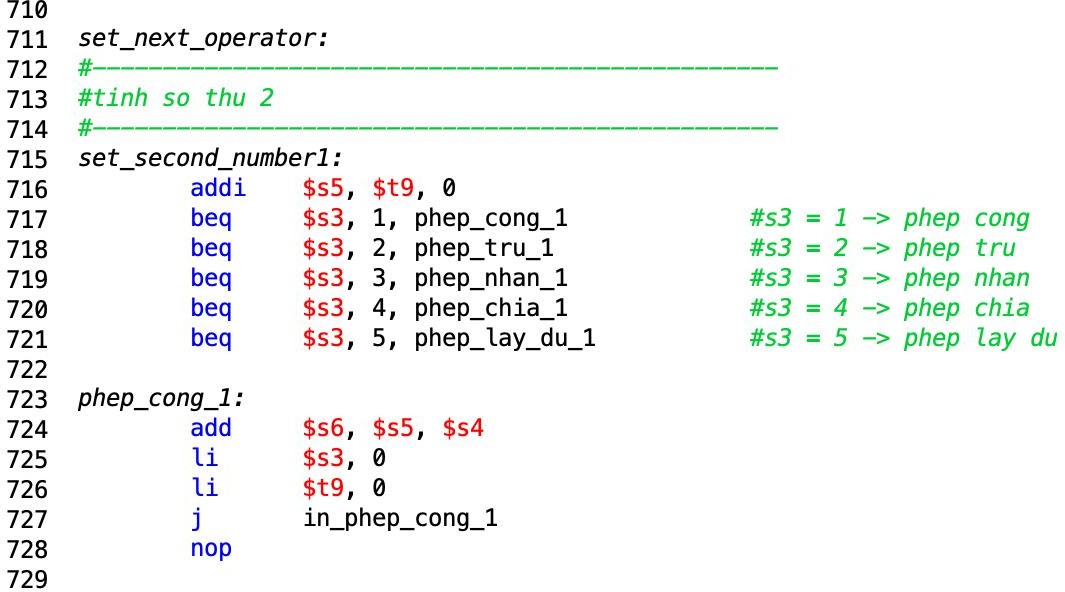


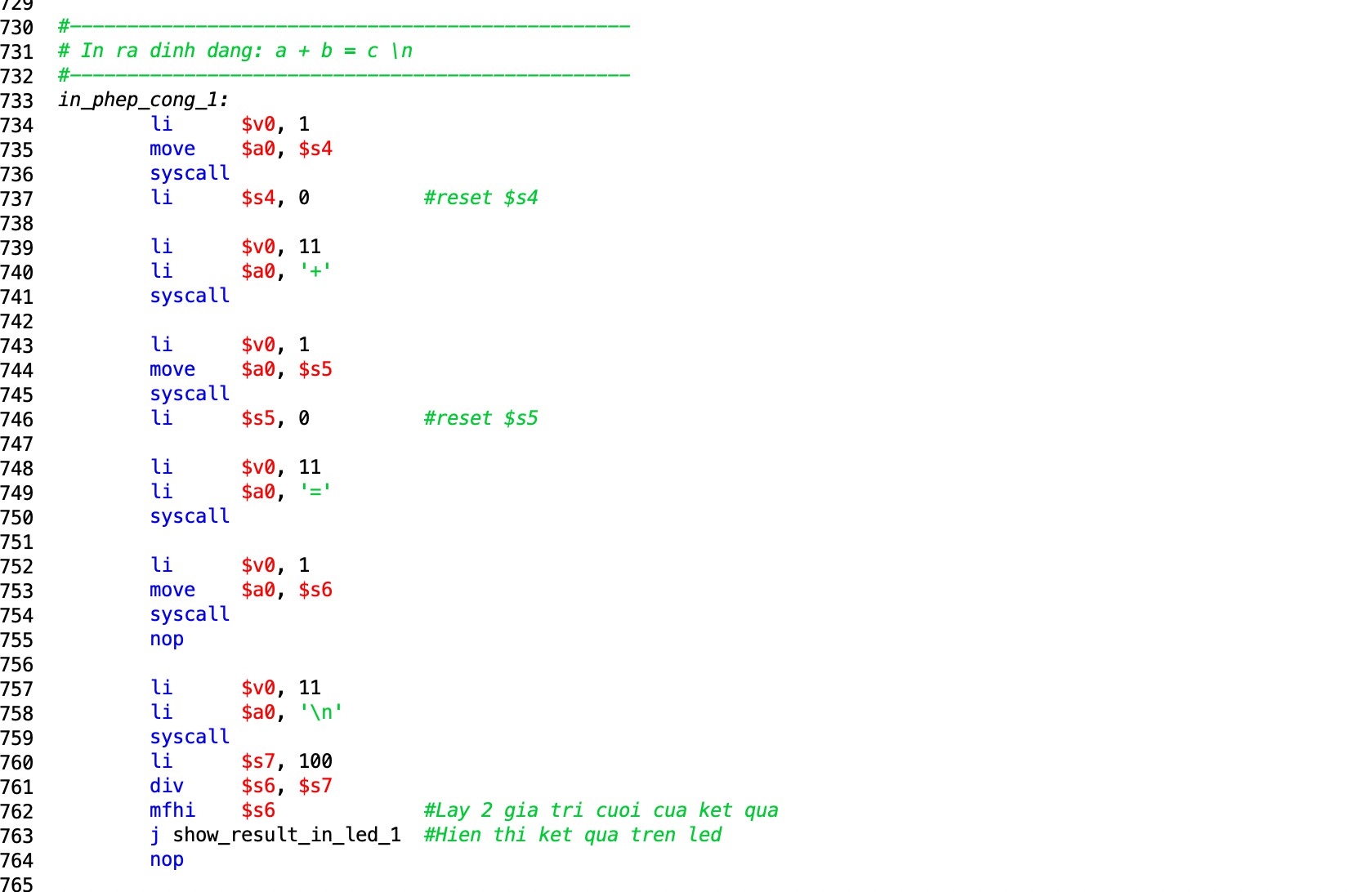




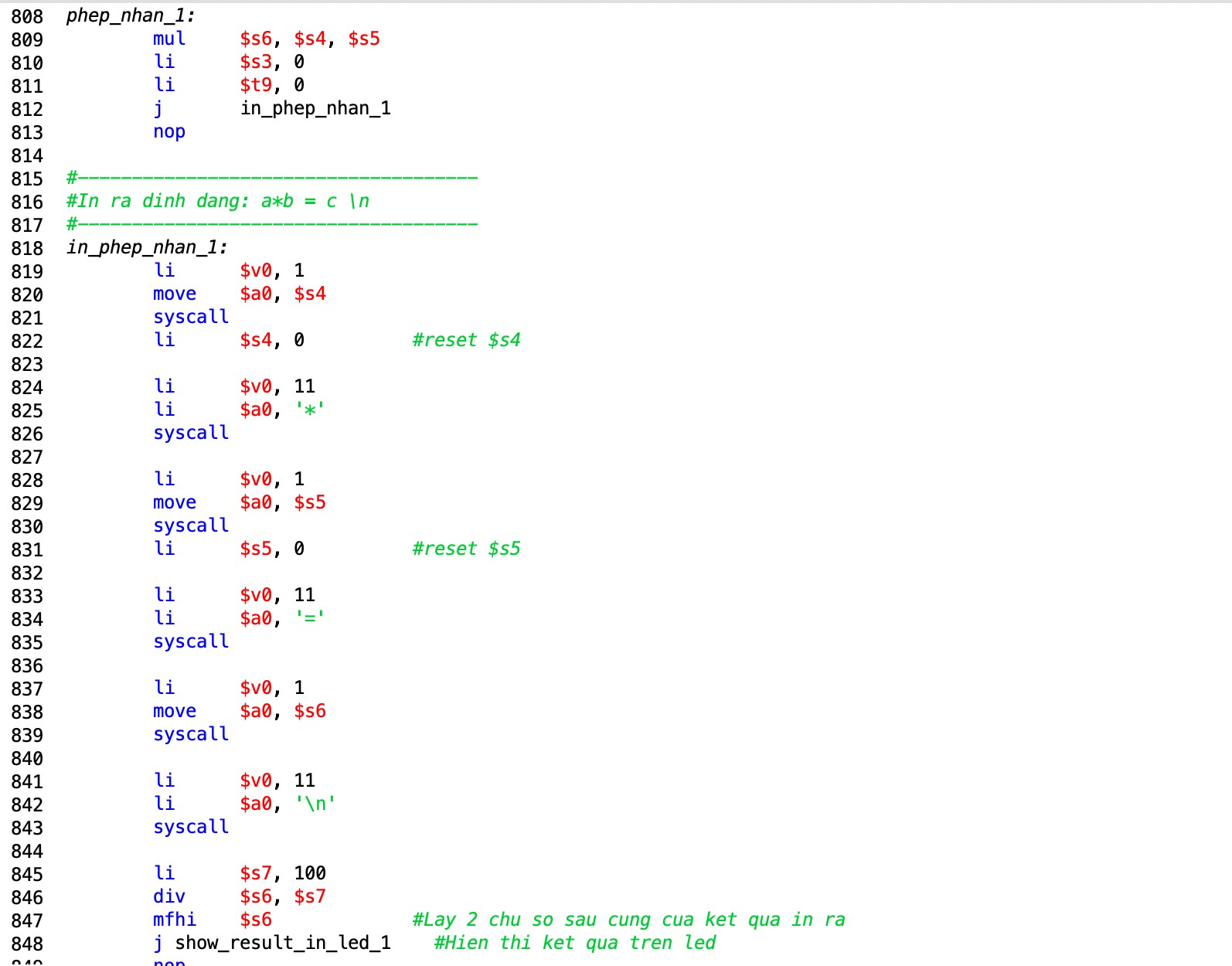


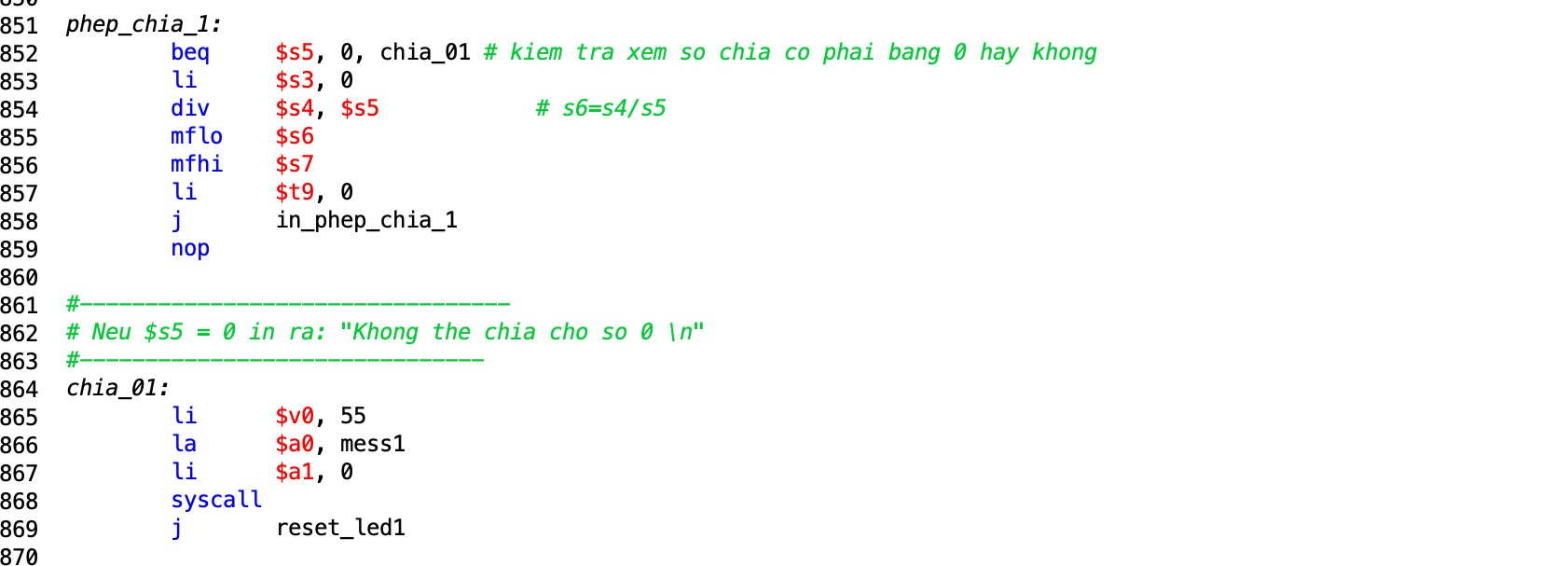


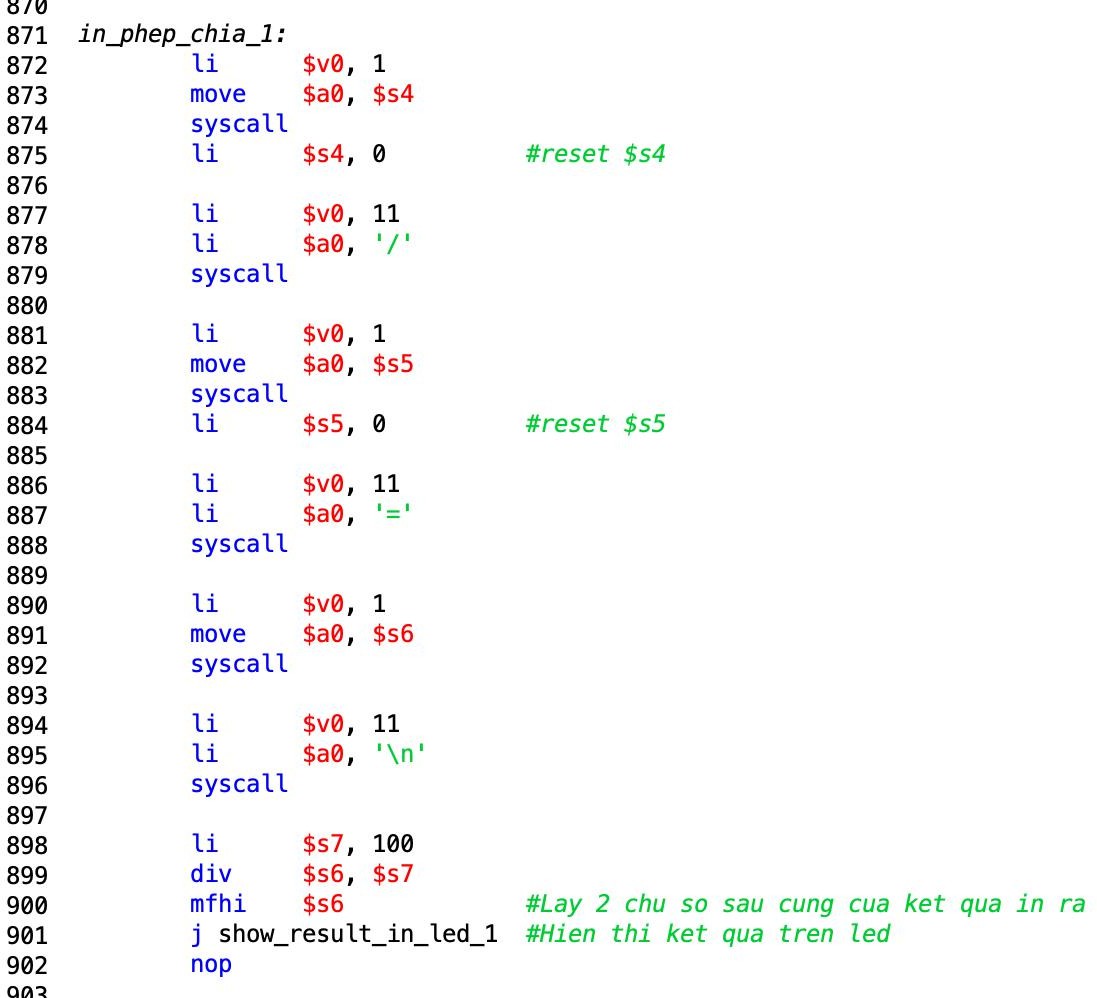


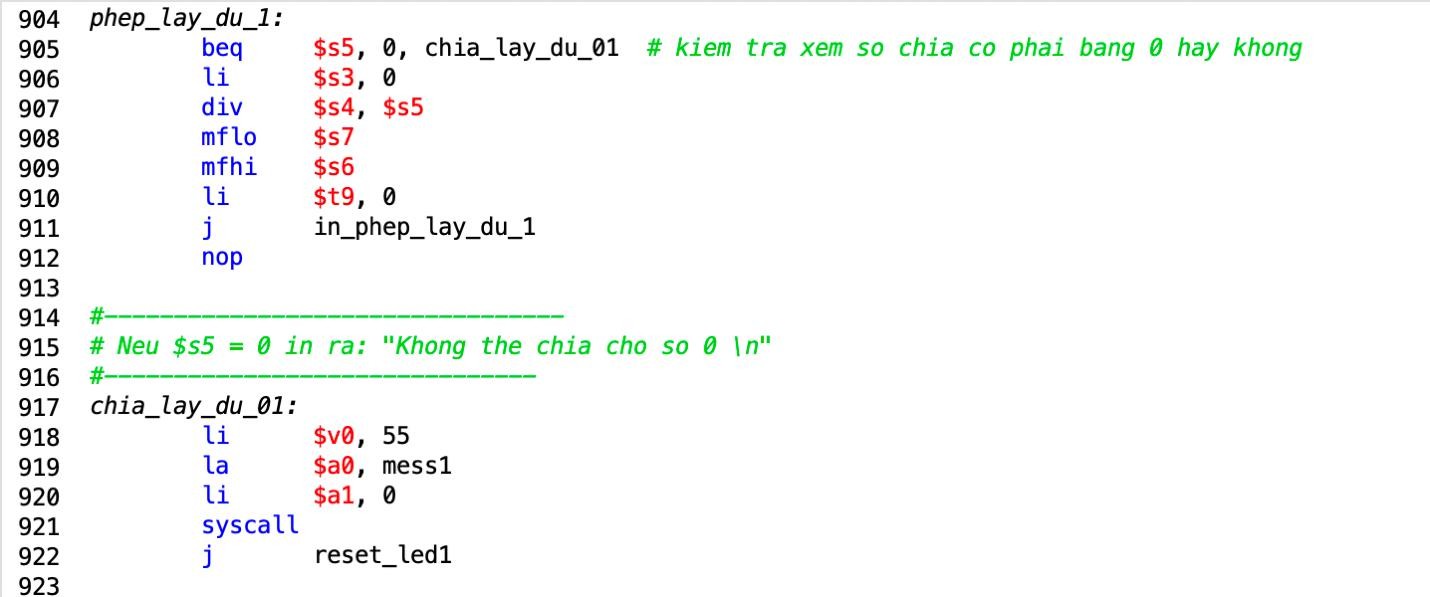


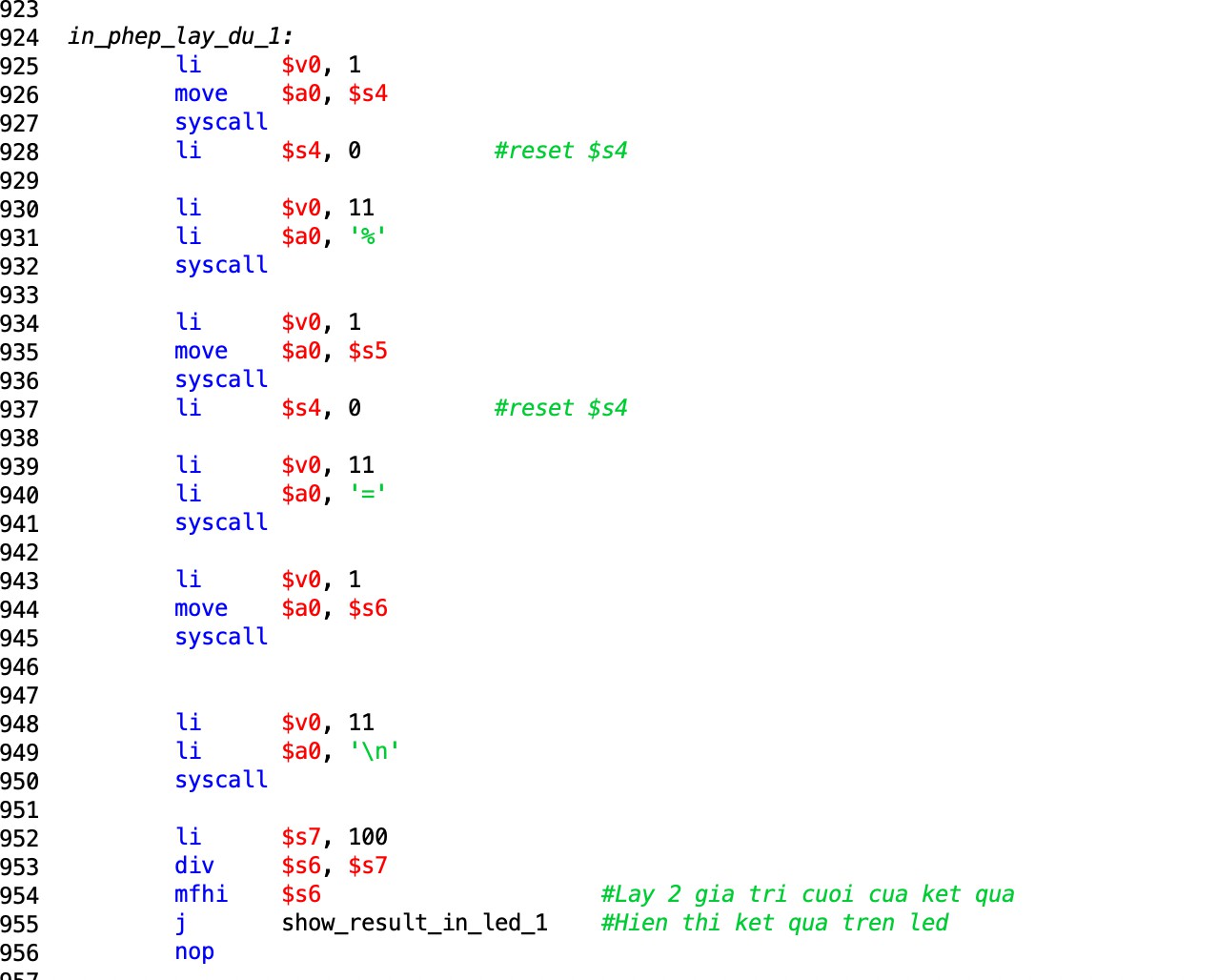


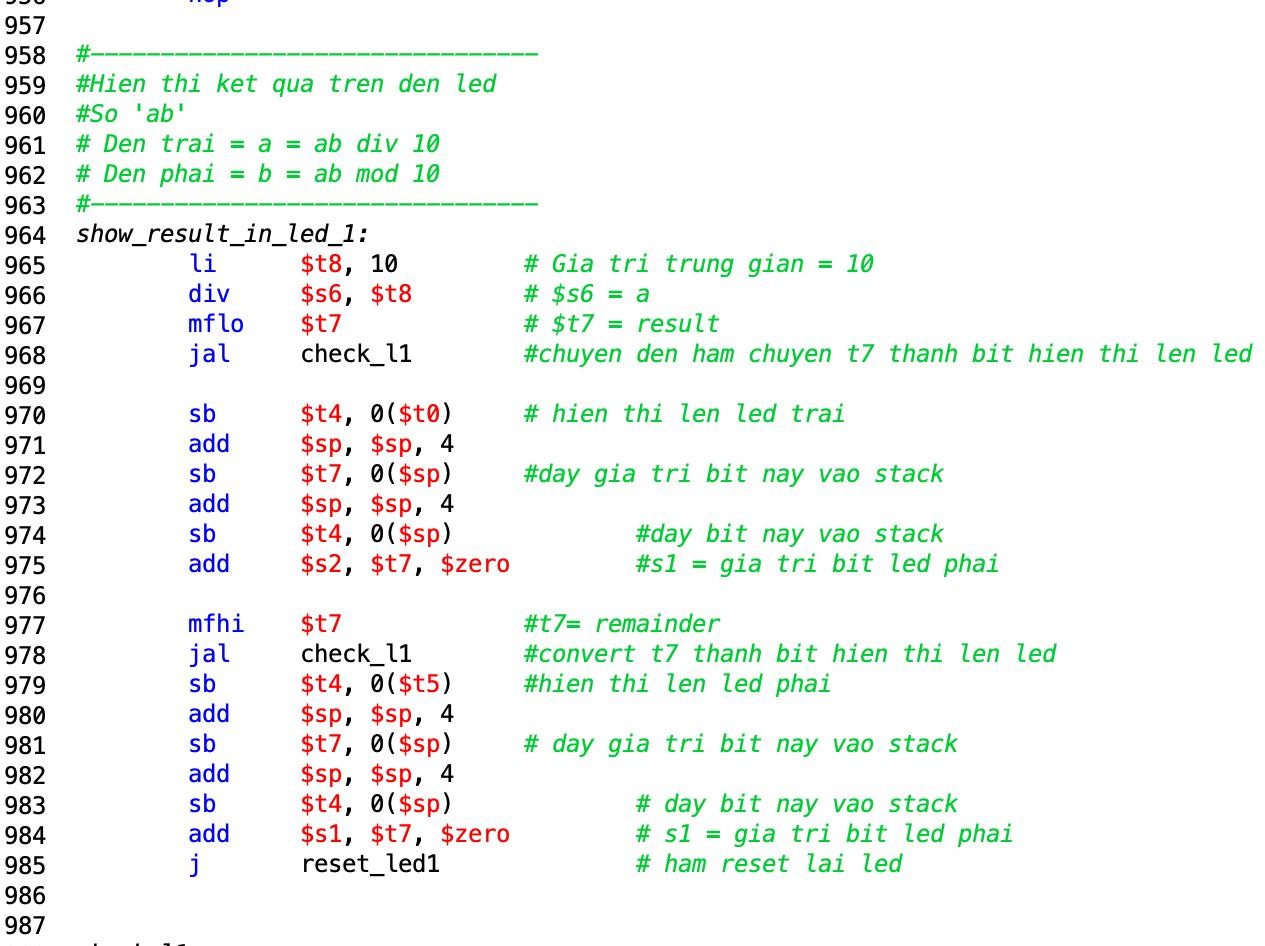


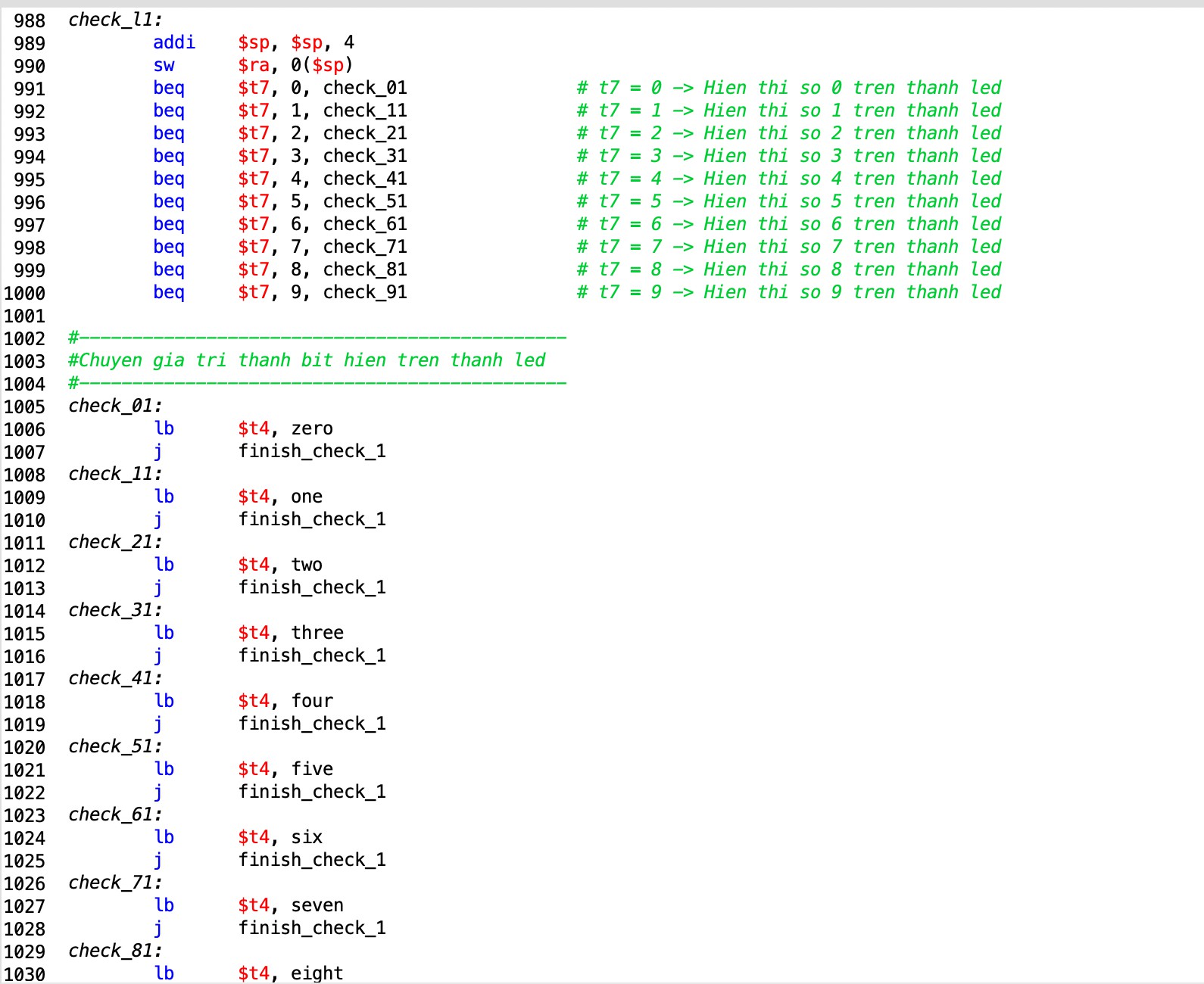


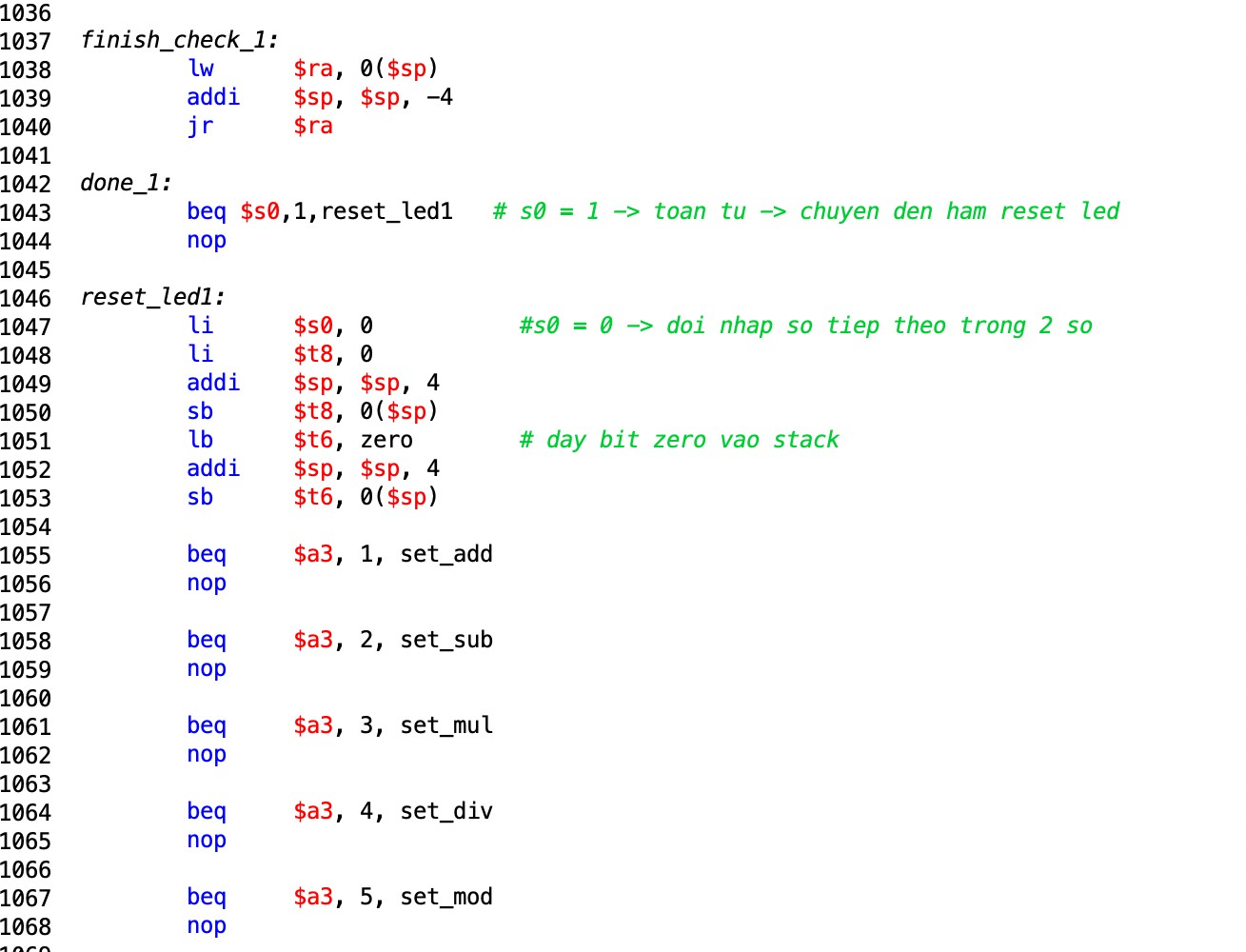


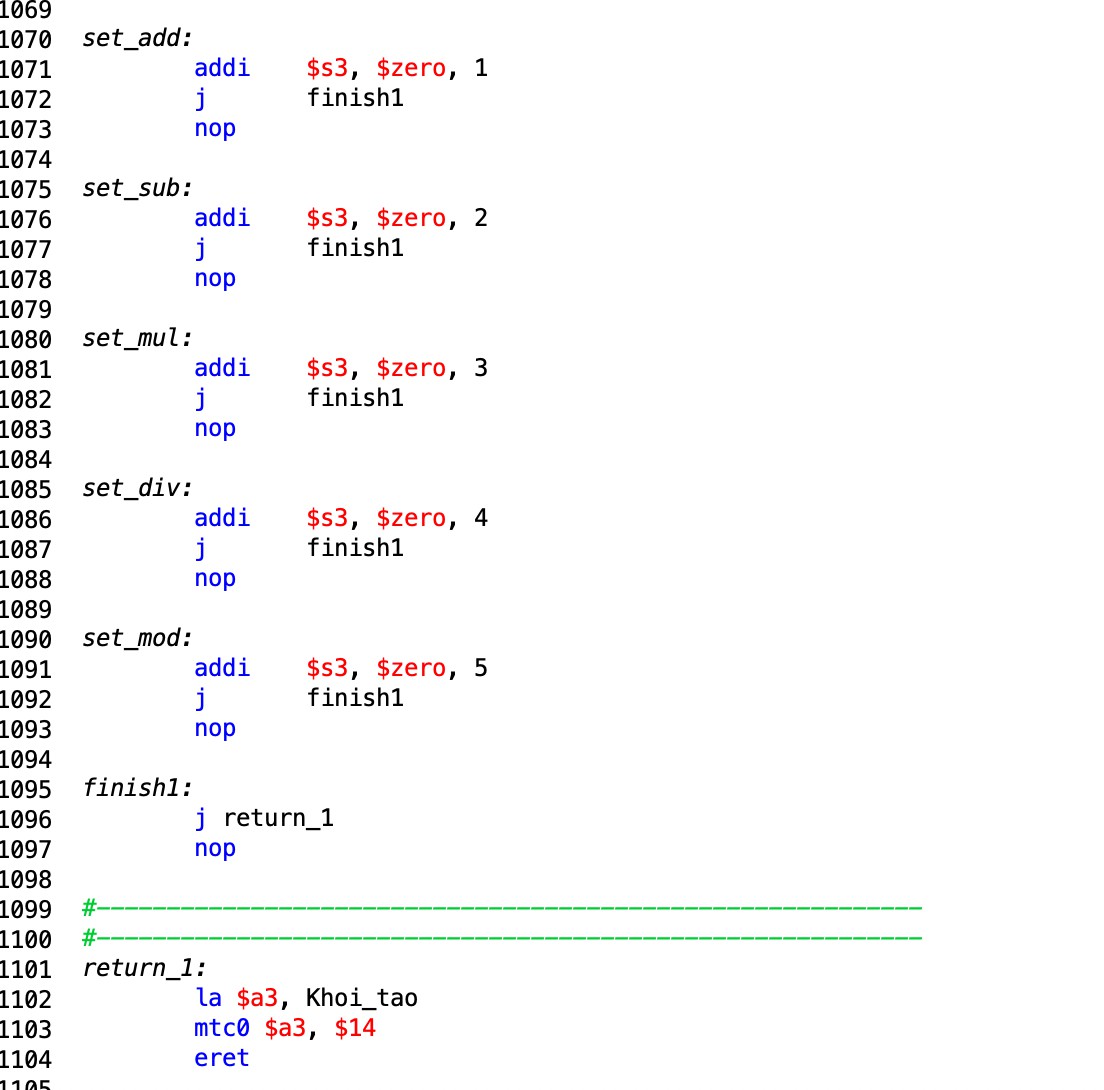










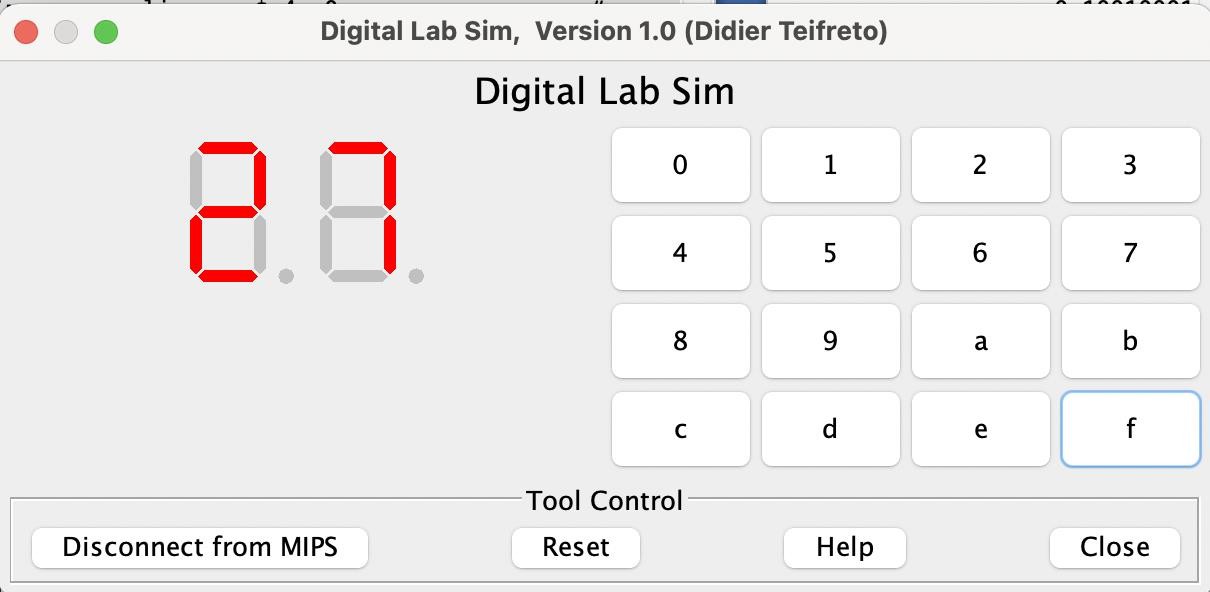


* 1. **Chạy thử chương trình Thực hiện các phép toán:**

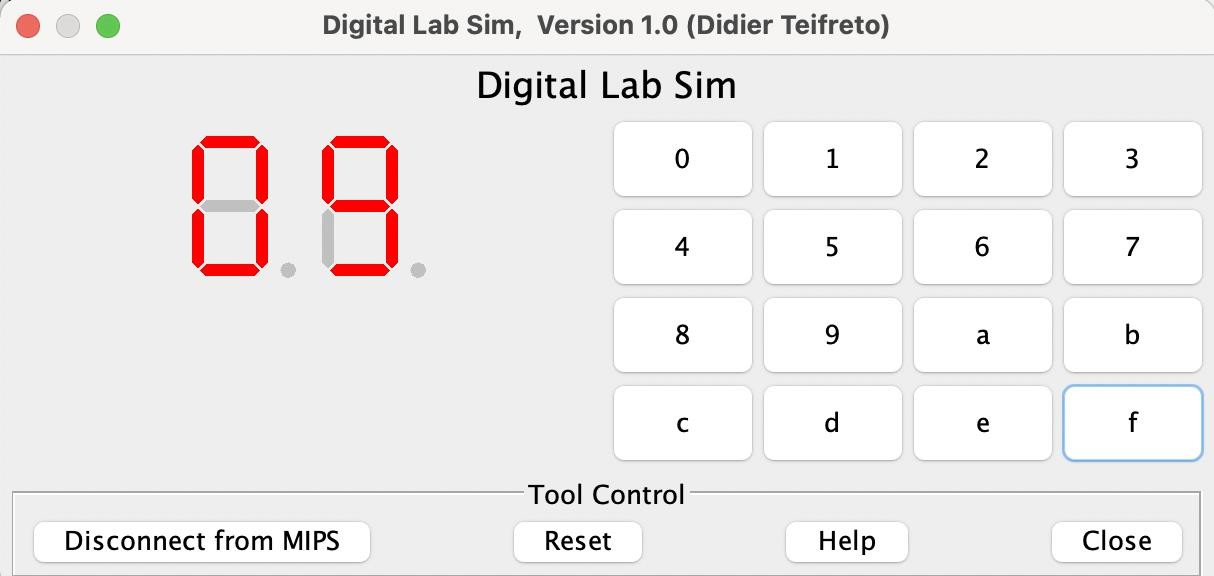


Hiển thị ra ngoài màn hình:

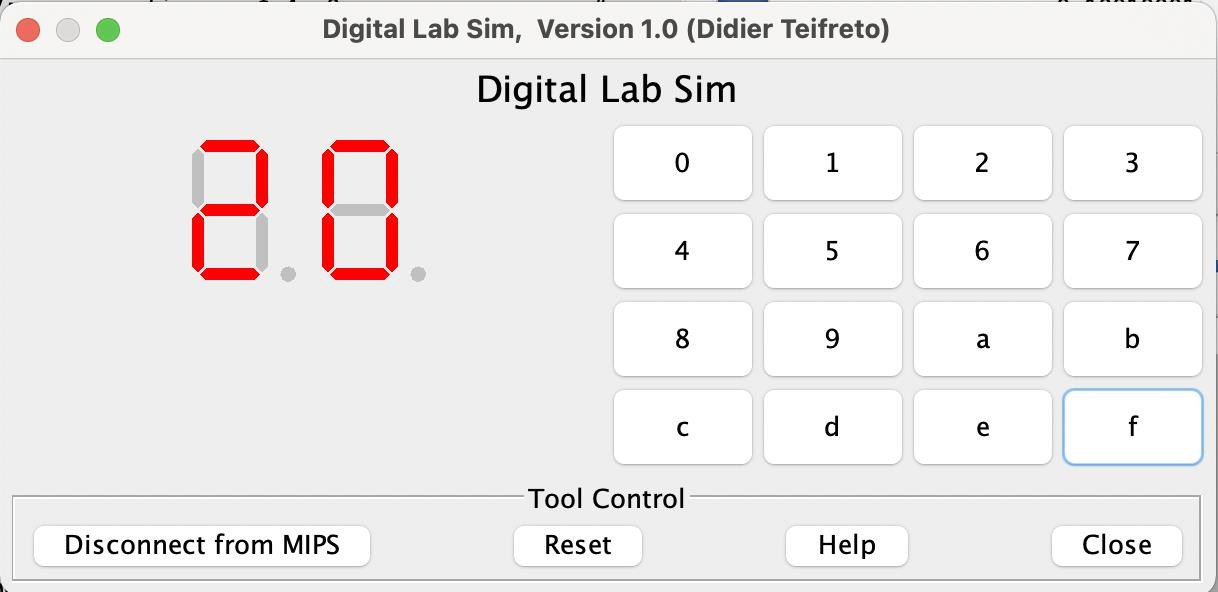
**15+112 = 127**



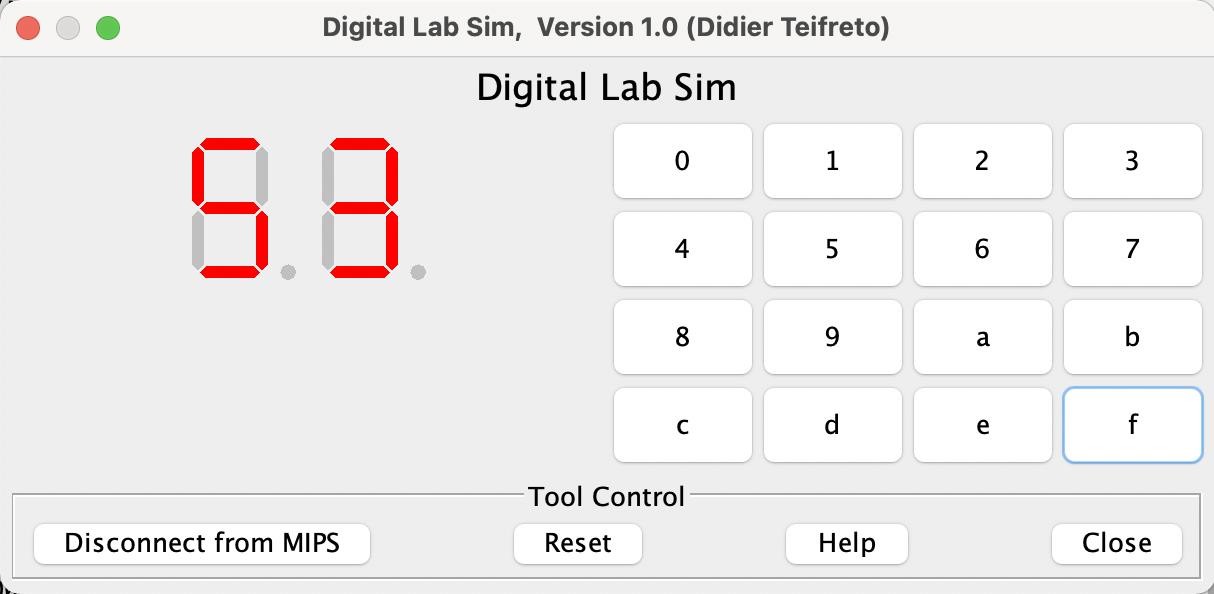
**15 – 6 = 9**



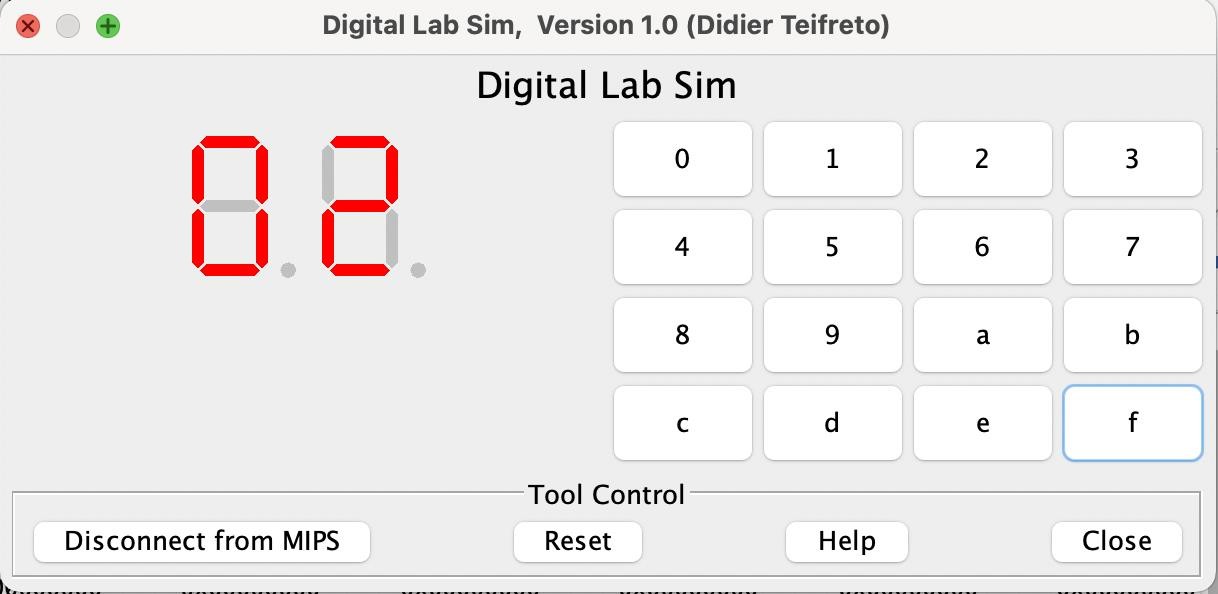
**15 \* 8 = 120**



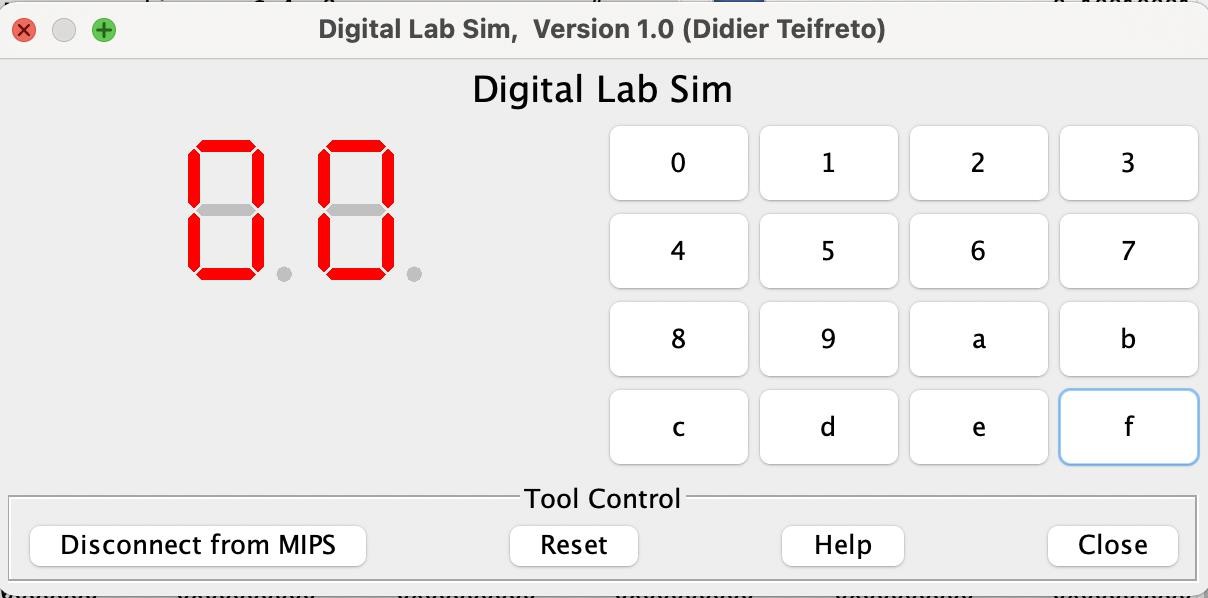
**159 / 3 = 53**



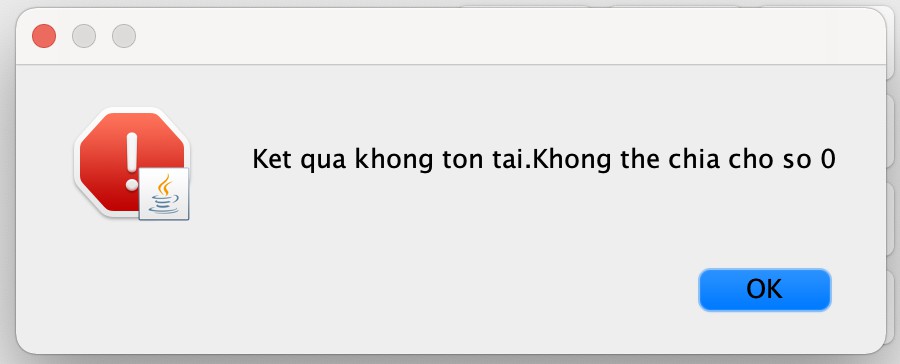
**17 % 3 = 2**



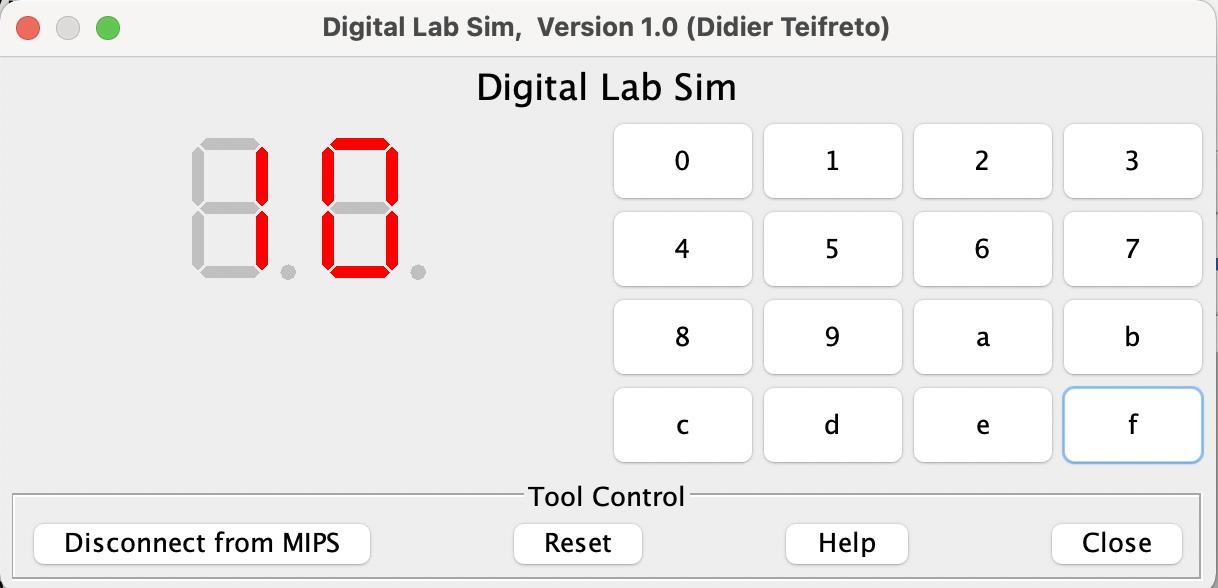
**12 – 45 = -33 -> không in được kết quả trên Digital Lab Sim.**



**51 / 0 -> Không thể thực hiện được.**



**51 / 5 = 10**



# 3. Project 3: Kiểm tra tốc độ và độ chính xác khi gõ văn bản

# a) phân tích cách làm , thuật toán

# - Sử dụng vòng lặp vô hạn. Trong vòng lặp có kiểm tra giá trị tại địa chỉ KEY\_READY nếu khác 0 thì có kí tự nhập từ bàn phím. Nếu có kí tự nhập từ bàn phím sẽ nhảy đến trạng thái MakeIntR để xử lý trạng thái ngắt.

# - Sau đó ta kiểm tra xem đó là loại ngắt nào bằng cách dùng phép AND :

# Nếu nó là ngắt từ bộ đếm thì sẽ thực hiện chương trình đếm số lần ngắt nếu số lần ngắt lớn hơn 40 ( tức là 1s) thì sẽ làm mới số lần ngắt và cộng thêm 1 giây vào biến đếm giây đã khai báo

# Nếu ngắt từ bàn phím thì ta sẽ thực hiện so sánh kí tự nhập vào với kí tự trong đoạn văn bản nếu giống nhau thì sẽ tăng biến đếm số kí tự đúng thêm 1 còn không giống nhau thì ta bỏ qua.

# - Sau đó quay trở về chương trình chính, nếu so sánh hết các ký tự trong đoạn văn bản hoặc khi người dùng ấn nút enter thì sẽ dừng vòng lặp

# - Cuối cùng ta in ra kết quả :

# Để in ra số ký tự đúng ra đèn LED ta dùng biến lưu số ký tự đúng chuyển sang 1 số khác để các thanh LED in ra đúng số

# Để tính số ký tự trên 1 đơn vị thời gian : ta lấy số ký tự được nhập vào nhân với 60 rồi chia cho thời gian nhập ký tự sẽ ra được số ký tự trên 1 phút

# b) Mã Nguồn

# *.eqv SEVENSEG\_LEFT 0xFFFF0011* *#dia chi led 7 doan trai*

# *.eqv SEVENSEG\_RIGHT 0xFFFF0010* *#dia chi led 7 doan phai*

# *.eqv MASK\_CAUSE\_COUNTER 0x00000400* *#Bit 10: Counter interrupt*

# *.eqv COUNTER 0xFFFF0013* *#Time Counter*

# *.eqv KEY\_CODE 0xFFFF0004* *# ASCII code from keyboard, 1 byte*

# *.eqv KEY\_READY 0xFFFF0000* *# =1 if has a new keycode ?*

# *# Auto clear after lw*

# *.data*

# *arr: .byte* *63, 6, 91, 79, 102, 109 ,125, 7, 127, 111* *#tu 0 den 9 de doi sang lam den led sang*

# *str: .asciiz "bo mon ki thuat may tinh"*

# *msg1: .asciiz "thoi gian hoan thanh : "*

# *msg2: .asciiz " (s)\nso tu tren mot don vi thoi gian : "*

# *msg3: .asciiz " tu/phut\n"*

# 

# 

# *#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

# *# MAIN Procedure*

# *#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

# 

# *.text*

# 

# *main:*

# *li $k0,KEY\_CODE*

# *li $k1,KEY\_READY*

# *# Enable the interrupt of TimeCounter of Digital Lab Sim*

# *li $t1, COUNTER*

# *sb $t1, 0($t1)*

# *addi $s0,$zero,0 # $s0 dem so ky tu nhap vao*

# *addi $s1,$zero,0 # $s1 dem tong so ky tu dung*

# *addi $s2,$zero,0 # dem so lan ngat*

# *addi $s3,$zero,0 # dem thoi gian*

# *la $a1, str*

# *#---------------------------------------------------------*

# *# nhap dau vao*

# *#---------------------------------------------------------*

# 

# *loop:*

# *lw $t1,0($k1)* *# $t1 = [$k1] = KEY\_READY*

# *bne $t1,0,MakeIntR* *# $t1 != 0 hay co phim duoc nhap thi chuyen den MakeIntR*

# *li $v0,32* *# BUG: must sleep to wait for Time*

# *li $a0,5* *# sleep 5ms*

# *syscall*

# *j loop*

# *MakeIntR:*

# *teqi $t1,1* *# neu $t1=1 co phim duoc nhap thi chuyen den chuong trinh ngat*

# *j loop* *# quay lai vong lap de nhap ki tu moi*

# 

# 

# 

# *#---------------------------------------------------------------*

# *# Interrupt subroutine*

# *#---------------------------------------------------------------*

# 

# *.ktext 0x80000180*

# *IntSR: #--------------------------------------------------------*

# *# Temporary disable interrupt*

# *#--------------------------------------------------------*

# 

# *dis\_int:li $t1, COUNTER # BUG: must disable with Time Counter*

# *sb $zero, 0($t1)*

# *#--------------------------------------------------------*

# *# Processing*

# *#--------------------------------------------------------*

# 

# *# kiem tra ly do ngat*

# 

# *get\_caus:*

# *mfc0 $t1,$13* *# $t1 la ly do ngat*

# 

# *IsCount:*

# *li $t2, MASK\_CAUSE\_COUNTER*

# *and $at,$t2,$t1*

# *bne $at,$t2,Keyboard\_IntR*

# 

# *# ly do ngat la bo dem*

# 

# *Counter\_IntR:*

# *ble $s2,40,demtiep* *# neu chua du 1s thi dem tiep*

# *addi $s2,$zero,0* *# tao lai so lan ngat*

# *addi $s3,$s3,1* *# tang len 1s*

# *j en\_int* *# nhay den xu ly chuoi*

# *demtiep:*

# *addi $s2,$s2,1*

# *j en\_int* *# nhay den xu ly chuoi*

# 

# *# xu ly chuoi*

# 

# *Keyboard\_IntR:*

# *lb $t0,0($a1)* *# lay ki tu thi i trong doan van ban*

# *beq $t0,$0,end\_program*

# *lw $t1,0($k0)* *# lay ki tu nhap vao tu ban phim*

# *bne $t0,$t1,end\_process # neu 2 ki tu ko bang nhau thi nhay den end\_process*

# *nop*

# *addi $s1,$s1,1* *# tang so ki tu dung len 1*

# *end\_process:*

# *addi $s0,$s0,1* *# tang so ki tu nhap vao len 1*

# *addi $a1,$a1,1* *# tang i 1 don vi trong doan van ban*

# 

# *en\_int:*

# *# Must clear cause reg*

# *#--------------------------------------------------------*

# *# Re-enable interrupt*

# *#--------------------------------------------------------*

# *li $t1, COUNTER*

# *sb $t1, 0($t1)*

# *mtc0 $zero, $13*

# *#--------------------------------------------------------*

# *# Evaluate the return address of main routine*

# *# epc <= epc + 4*

# *#--------------------------------------------------------*

# *next\_pc:mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc*

# *addi $at, $at, 4 # $at = $at + 4 (next instruction)*

# *mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at*

# *return: eret # Return from exception*

# *#-----------------------------------------------------------------*

# 

# *end\_program:*

# *# in ra thoi gian hoan thanh*

# *addi* *$v0, $0, 4*

# *la* *$a0, msg1*

# *syscall*

# *addi* *$v0, $0, 1*

# *addi* *$a0, $s3, 0*

# *syscall*

# *addi* *$v0, $0, 4*

# *la* *$a0, msg2*

# *syscall*

# *# in ra so tu tren 1 don vi thoi gian*

# *addi* *$v0, $0, 1*

# *addi* *$a0, $0, 60*

# *mult* *$s0, $a0* *# nhan so ky tu nhap vao voi 60*

# *mflo* *$s0*

# *div* *$s0, $s3* *# so ky tu trong 1p*

# *mflo* *$a0*

# *syscall*

# *addi* *$v0, $0, 4*

# *la* *$a0, msg3*

# *syscall*

# *addi* *$s0, $s1, 0*

# *jal* *show*

# *end\_main:*

# *li $v0,10*

# *syscall*

# *#-----------------------------------------------------------------*

# *# hien thi ket qua so ky tu dung tren led*

# *show:*

# *addi $sp,$sp,-4*

# *sw $ra,0($sp)*

# *addi $t0,$zero,10*

# *div $s1,$t0*

# *mflo $v0* *# $v0 = so hang chuc cua so ky tu dung*

# *mfhi $v1* *# $v1 = so hang don vi cua so ky tu dung*

# *la $a0, arr*

# *add $a0, $a0, $v0* *# $a0 la so de hien thi ra $v0*

# *lb $a0, 0($a0)* *#Set value for segments*

# *jal SHOW\_7SEG\_LEFT*

# *la $a0, arr*

# *add $a0, $a0, $v1* *# $a0 la so de hien thi ra $v1*

# *lb $a0, 0($a0)* *#Set value for segments*

# *jal SHOW\_7SEG\_RIGHT*

# *lw $ra, ($sp)*

# *addi $sp, $sp, 4*

# *jr $ra*

# *SHOW\_7SEG\_LEFT:*

# *li* *$t0, SEVENSEG\_LEFT* *#Assign port's address*

# *sb* *$a0, 0($t0)* *#Assign new value*

# *jr* *$ra*

# *SHOW\_7SEG\_RIGHT:*

# *li* *$t0, SEVENSEG\_RIGHT* *#Assign port's address*

# *sb* *$a0, 0($t0)* *#Assign new value*

# *jr* *$ra*

# *nop*

# 

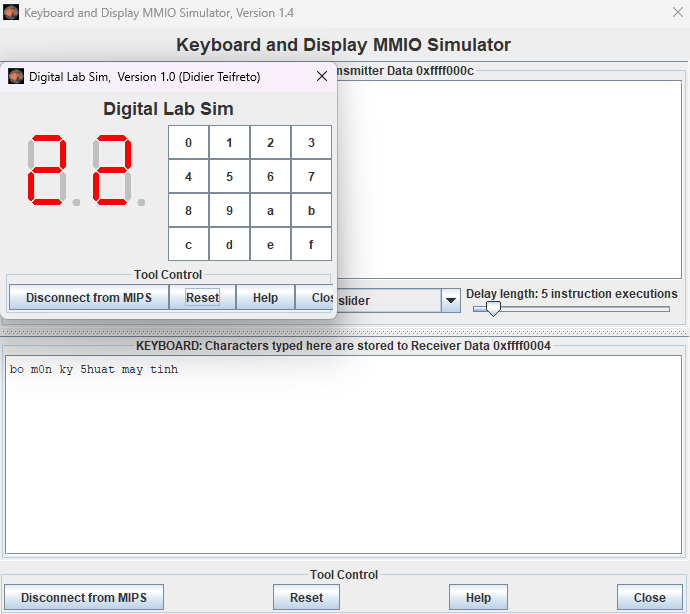
# *#------------------------------------------------------------------------*

# 

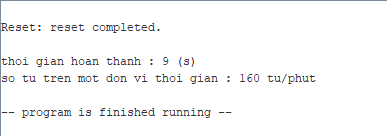
# 

# **c) Kết quả chạy mô phỏng**

Khi ta nhập “bo m0n ki 5huat may tinh”



Nhập sai chữ o và t nên sẽ là 22



Vì số từ nhập là 24 và thời gian chạy nên số từ trên 1 phút là 160

## BÀI BÁO CÁO ĐẾN ĐÂY LÀ KẾT THÚC

  

**Cảm ơn thầy đã đọc, chúc thầy một ngày làm việc hiệu quả!**