



BÁO CÁO CUỐI KỲ

Bộ môn: Thực Hành Kiến Trúc Máy Tính



GIÁO VIÊN BỘ MÔN: LÊ BÁ VUI

NHÓM 6 - LỚP 130937

Nguyễn Bá Hoàng 20204976

Phạm Vũ Hoàng 20208000

I. Bài 2: Vẽ hình trên màn hình bitmap

1. Phân tích cách làm

Bài toán được chia làm các phần cơ bản như sau:

- B1: Vẽ đường tròn ban đầu
- B2: Nhập phím điều khiển (có thể không nhập)
- B3: Vẽ lại đường tròn dựa theo hướng di chuyển hiện tại đồng thời xóa đường tròn cũ
- B4: Quay lại bước 2

2. Thuật toán

Chi tiết các bước:

B1: sử dụng hàm draw với đầu vào hàm draw là địa chỉ chính giữa bitmap (tâm đường tròn)

B2: nhận phím từ keyboard, nếu có phím nhập vào thì thay đổi hướng di chuyển hoặc tốc độ vẽ (thay đổi thời gian sleep)

B3_0: cho chương trình sleep 1 khoảng thời gian cố định (đủ nhỏ)

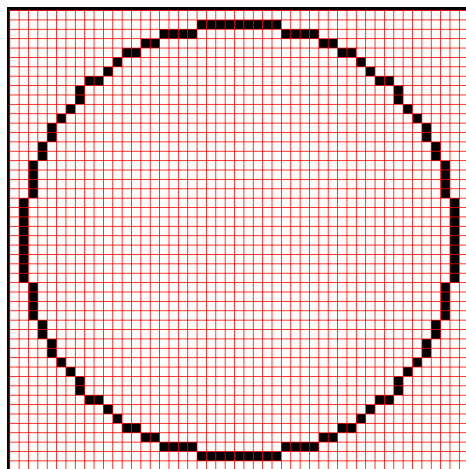
B3a: xóa đường tròn cũ bằng cách gọi hàm draw với màu vẽ là màu đen với đầu vào là địa chỉ tâm đường tròn trước khi có sự thay đổi địa chỉ

B3b: thay đổi địa chỉ tâm đường tròn dựa theo hướng di chuyển hiện tại (tăng hoặc giảm) (nếu đạt đến các giới hạn, nghĩa là đường tròn va vào các cạnh của bitmap thì undo việc thay đổi địa chỉ tâm đường tròn và đảo ngược lại hướng di chuyển của tâm (như được nhập phím từ keyboard) quay lại bước 2)

B3c: vẽ đường tròn mới dựa theo địa chỉ mới (gọi hàm draw với đầu vào là địa chỉ tâm đường tròn mới)

B4: quay lại bước 2

- Hàm draw vẽ từng điểm trên đường tròn bằng 132 câu lệnh lưu mã màu vào các địa chỉ tương ứng (sử dụng thanh ghi \$t0 (mã màu) và \$s1 (địa chỉ tâm đường tròn))



Hình tròn đã sử dụng để tính toán địa chỉ

3. Mã nguồn

```
.eqv    screen      0x10010000
.eqv    yellow_color 0x00FFFF00
.eqv    black_color  0x00000000
.eqv    KEY_CODE     0xFFFF0004
.eqv    KEY_READY    0xFFFF0000
.text
li      $k0, KEY_CODE
li      $k1, KEY_READY
li      $s0,screen
li      $t0,yellow_color
li      $a2,50        #thoi gian sleeping mac dinh(toc do)
li      $s2,0         #huong di chuyen cua dia chi
li      $s3,0         #huong di chuyen ngang
li      $s4,0         #huong di chuyen doc
li      $t1,255       #toa do x
li      $t2,254       #toa do y
li      $t3,23        #can tren va can trai toa do tam
li      $t4,488       #can duoi va can phai toa do tam
addi    $s1,$s0,521212 #dia chi ban dau cua tam hình tron
jal     draw
j       loop
Change:                                #da co phim duoc bam tu keyboard
lw      $v0, 0($k0)
li      $v1,'w'
beq     $v1,$v0,up
li      $v1,'s'
beq     $v1,$v0,down
li      $v1,'d'
```

```

beq    $v1,$v0,right
li     $v1,'a'
beq    $v1,$v0,left
li     $v1,'z'
beq    $v1,$v0,speed_up
li     $v1,'x'
beq    $v1,$v0,slow_down
j      loop
up:
                                #giam dia chi tam de tang duoc 1 hang
li     $s2,-2048
li     $s3,0
li     $s4,-1
j      loop
down:
                                #tang dia chi tam de xuong duoc 1 hang
li     $s2,2048
li     $s3,0
li     $s4,1
j      loop
right:
                                #tang dia chi tam de sang phai duoc 1 cot
li     $s2,4
li     $s3,1
li     $s4,0
j      loop
left:
                                #giam dia chi tam de sang trai duoc 1 cot
li     $s2,-4
li     $s3,-1
li     $s4,0
j      loop
speed_up:
                                #giam thoi gian sleep giua cac vong lap de tang toc

```

```

addi    $a2,$a2,-5
j        loop

slow_down:
                                #tang thoi gian sleep giua cac vong lap de giam toc do

addi    $a2,$a2,5
j        loop

undo_right:
                                #reset toa do tam ve buoc truoc(vi da ra qua gioi han cho phep)

sub     $s1,$s1,$s2            #thay doi dia chi tam duong tron
sub     $t1,$t1,$s3            #thay doi toa do x
sub     $t2,$t2,$s4            #thay doi toa do y

j        right

undo_left:

sub     $s1,$s1,$s2            #thay doi dia chi tam duong tron
sub     $t1,$t1,$s3            #thay doi toa do x
sub     $t2,$t2,$s4            #thay doi toa do y

j        left

undo_down:

sub     $s1,$s1,$s2            #thay doi dia chi tam duong tron
sub     $t1,$t1,$s3            #thay doi toa do x
sub     $t2,$t2,$s4            #thay doi toa do y

j        down

undo_up:

sub     $s1,$s1,$s2            #thay doi dia chi tam duong tron
sub     $t1,$t1,$s3            #thay doi toa do x
sub     $t2,$t2,$s4            #thay doi toa do y

j        up

loop:

addi    $v0,$zero,32           #sleep

add     $a0,$zero,$a2

syscall

```

```

li      $t0,black_color      #xoa duong tron cu
jal     draw
add     $s1,$s1,$s2          #thay doi dia chi tam duong tron
add     $t1,$t1,$s3          #thay doi toa do x
add     $t2,$t2,$s4          #thay doi toa do y
beq     $t3,$t1,undo_right   #va cham canh trai
beq     $t3,$t2,undo_down    #va cham canh tren
beq     $t4,$t1,undo_left    #va cham canh phai
beq     $t4,$t2,undo_up      #va cham canh duoi
li      $t0,yellow_color     #ve duong tron moi
jal     draw
lw      $a0, 0($k1)          #loop doi keyboard
beq     $a0, $zero, loop
j       Change
draw:                                       #ve duong tron gom 132 pixel
sw      $t0,-47120($s1)
sw      $t0,-47116($s1)
sw      $t0,-47112($s1)
sw      $t0,-47108($s1)
sw      $t0,-47104($s1)
sw      $t0,-47100($s1)
sw      $t0,-47096($s1)
sw      $t0,-47092($s1)
sw      $t0,-47088($s1)
sw      $t0,-45076($s1)
sw      $t0,-45080($s1)
sw      $t0,-45084($s1)
sw      $t0,-45088($s1)
sw      $t0,-45036($s1)

```

sw	\$t0,-45032(\$s1)
sw	\$t0,-45028(\$s1)
sw	\$t0,-45024(\$s1)
sw	\$t0,-43044(\$s1)
sw	\$t0,-43048(\$s1)
sw	\$t0,-42972(\$s1)
sw	\$t0,-42968(\$s1)
sw	\$t0,-41004(\$s1)
sw	\$t0,-41008(\$s1)
sw	\$t0,-40916(\$s1)
sw	\$t0,-40912(\$s1)
sw	\$t0,-38964(\$s1)
sw	\$t0,-38860(\$s1)
sw	\$t0,-36920(\$s1)
sw	\$t0,-36808(\$s1)
sw	\$t0,-34876(\$s1)
sw	\$t0,-34880(\$s1)
sw	\$t0,-34756(\$s1)
sw	\$t0,-34752(\$s1)
sw	\$t0,-32836(\$s1)
sw	\$t0,-32700(\$s1)
sw	\$t0,-30788(\$s1)
sw	\$t0,-30652(\$s1)
sw	\$t0,-28744(\$s1)
sw	\$t0,-28600(\$s1)
sw	\$t0,-26700(\$s1)
sw	\$t0,-26548(\$s1)
sw	\$t0,-24656(\$s1)
sw	\$t0,-24496(\$s1)

sw	\$t0,-22608(\$s1)
sw	\$t0,-22448(\$s1)
sw	\$t0,-20564(\$s1)
sw	\$t0,-20396(\$s1)
sw	\$t0,-18516(\$s1)
sw	\$t0,-18348(\$s1)
sw	\$t0,-16472(\$s1)
sw	\$t0,-16296(\$s1)
sw	\$t0,-14424(\$s1)
sw	\$t0,-14248(\$s1)
sw	\$t0,-12376(\$s1)
sw	\$t0,-12200(\$s1)
sw	\$t0,-10328(\$s1)
sw	\$t0,-10152(\$s1)
sw	\$t0,-8284(\$s1)
sw	\$t0,-8100(\$s1)
sw	\$t0,-6236(\$s1)
sw	\$t0,-6052(\$s1)
sw	\$t0,-4188(\$s1)
sw	\$t0,-4004(\$s1)
sw	\$t0,-2140(\$s1)
sw	\$t0,-1956(\$s1)
sw	\$t0,-92(\$s1)
sw	\$t0,47120(\$s1)
sw	\$t0,47116(\$s1)
sw	\$t0,47112(\$s1)
sw	\$t0,47108(\$s1)
sw	\$t0,47104(\$s1)
sw	\$t0,47100(\$s1)

sw	\$t0,47096(\$s1)
sw	\$t0,47092(\$s1)
sw	\$t0,47088(\$s1)
sw	\$t0,45076(\$s1)
sw	\$t0,45080(\$s1)
sw	\$t0,45084(\$s1)
sw	\$t0,45088(\$s1)
sw	\$t0,45036(\$s1)
sw	\$t0,45032(\$s1)
sw	\$t0,45028(\$s1)
sw	\$t0,45024(\$s1)
sw	\$t0,43044(\$s1)
sw	\$t0,43048(\$s1)
sw	\$t0,42972(\$s1)
sw	\$t0,42968(\$s1)
sw	\$t0,41004(\$s1)
sw	\$t0,41008(\$s1)
sw	\$t0,40916(\$s1)
sw	\$t0,40912(\$s1)
sw	\$t0,38964(\$s1)
sw	\$t0,38860(\$s1)
sw	\$t0,36920(\$s1)
sw	\$t0,36808(\$s1)
sw	\$t0,34876(\$s1)
sw	\$t0,34880(\$s1)
sw	\$t0,34756(\$s1)
sw	\$t0,34752(\$s1)
sw	\$t0,32836(\$s1)
sw	\$t0,32700(\$s1)

sw	\$t0,30788(\$s1)
sw	\$t0,30652(\$s1)
sw	\$t0,28744(\$s1)
sw	\$t0,28600(\$s1)
sw	\$t0,26700(\$s1)
sw	\$t0,26548(\$s1)
sw	\$t0,24656(\$s1)
sw	\$t0,24496(\$s1)
sw	\$t0,22608(\$s1)
sw	\$t0,22448(\$s1)
sw	\$t0,20564(\$s1)
sw	\$t0,20396(\$s1)
sw	\$t0,18516(\$s1)
sw	\$t0,18348(\$s1)
sw	\$t0,16472(\$s1)
sw	\$t0,16296(\$s1)
sw	\$t0,14424(\$s1)
sw	\$t0,14248(\$s1)
sw	\$t0,12376(\$s1)
sw	\$t0,12200(\$s1)
sw	\$t0,10328(\$s1)
sw	\$t0,10152(\$s1)
sw	\$t0,8284(\$s1)
sw	\$t0,8100(\$s1)
sw	\$t0,6236(\$s1)
sw	\$t0,6052(\$s1)
sw	\$t0,4188(\$s1)
sw	\$t0,4004(\$s1)
sw	\$t0,2140(\$s1)

```
sw    $t0,1956($s1)
sw    $t0,92($s1)
jr     $ra
```

II Bài 5: Biến đổi biểu thức trung tố sang hậu tố và tính toán

1. Phân tích cách làm

- Nhập vào chuỗi biểu thức trung tố
- Sắp xếp lại biểu thức trung tố sang hậu tố bằng thuật toán
- Tính toán kết quả dựa trên biểu thức hậu tố

2. Thuật toán (chi tiết các bước)

- Nhập vào chuỗi biểu thức trung tố rồi in ra chuỗi đó.

Input

×

?

Nhap bieu thuc trung to
Note: chi su dung + - * / % ()
Cac so tu 00-99

OK

Cancel

- Khởi tạo các giá trị như trạng thái, số chữ số, khoảng cách giữa đại chỉ top và địa chỉ đầu của operator và postfix, địa chỉ của infix, postfix và operator.

- Scan input cho đến hết (hết input thì pop hết ở operator sang postfix)

- Trong quá trình scan đồng thời kiểm tra các giá trị trong chuỗi xem có hợp lệ không và phân các số vào stack postfix và các toán tử vào stack operator.

- Giải mã các toán tử rồi pop sang stack postfix.
- In ra các kí tự trong stack postfix.
- Lấy ra các số và toán tử trong stack postfix ra để tính toán.
- +Check xem toán tử là gì để tính.
- In ra kết quả tính

```
Bieu thuc trung to: ( 9 - 5 ) * 2 - 7 % 4 + 2
Bieu thuc hau to: 9 5 - 2 * 7 4 % - 2 +
Ket qua: 7
```

*Các tính năng của chương trình con đã được comment trong code dưới đây:

.data

infix: .space 256

postfix: .space 256

operator: .space 256

endMsg: .ascii "Ban co muon tiep tục không?"

byeMsg: .ascii "Ket thuc chương trình"

errorMsg: .ascii "Input error"

startMsg: .ascii "Nhap bieu thuc trung to\nNote: chỉ sử dụng + - * / % ()\nCac số từ 00-99"

prompt_postfix: .ascii "Bieu thuc hau to: "

prompt_result: .ascii "Ket qua: "

prompt_infix: .ascii "Bieu thuc trung to: "

reset: .word 1

stack: .word

.text

start:

Nhap bieu thuc trung to

li \$v0, 54

la \$a0, startMsg

la \$a1, infix

la \$a2, 256

```

syscall
beq $a1,-2,end
beq $a1,-3,start
# In ra Trung to
li $v0, 4
la $a0, prompt_infix
syscall
li $v0, 4
la $a0, infix
syscall
li $v0, 11
li $a0, '\n'
syscall
# Trang thai
li $s7,0          # Trang thai
# 0 = ban dau khong nhan vao anything
# 1 = nhap vao so
# 2 = nhap vao toan tu
# 3 = nhap vao dau (
# 4 = nhapm vao dau )
li $t9,0          # Dem chu so
li $t5,-1         # Postfix top offset
li $t6,-1         # Operator top offset
la $t1, infix     # Infix dia chi byte hien tai +1 mỗi loop
la $t2, postfix
la $t3, operator
addi $t1,$t1,-1    # Dat dia chi ban dau cua infix ve -1
# Chuyen thanh hau to
scanInfix:        # Loop doi voi moi 1 ki tu trong postfix

```

```

# Kiem tra input
addi $t1,$t1,1          # Tang vi tri infix
lb $t4, ($t1)           # Load infix input hien tai
beq $t4, ' ', scanInfix  # Neu scan dau cach thi bo qua va scan lai
beq $t4, '\n', EOF       # Scan den het input --> pop moi operator sang postfix
beq $t9,0,digit1        # Neu trang thai la 0 chu so
beq $t9,1,digit2        # Neu trang thai la 1 chu so
beq $t9,2,digit3        # Neu trang thai la 2 chu so
continueScan:
beq $t4, '+', plusMinus
beq $t4, '-', plusMinus
beq $t4, '*', multiplyDivide
beq $t4, '/', multiplyDivide
beq $t4, '%', multiplyDivide
beq $t4, '(', openBracket
beq $t4, ')', closeBracket
wrongInput:    # Khi phat hien input nhap sai
li $v0, 55
la $a0, errorMsg
li $a1, 2
syscall
j ask
finishScan:
# In bieu thuc hau to
# Print prompt:
li $v0, 4
la $a0, prompt_postfix
syscall
li $t6,-1      # Load Postfix offset hien tai ve -1

```

```

printPost:
    addi $t6,$t6,1          # Tang Postfix offset hien tai
    add $t8,$t2,$t6         # Load dia chi cua Postfix hien tai
    lbu $t7,($t8)           # Load gia tri cua Postfix hien tai
    bgt $t6,$t5,finishPrint # In moi postfix --> tinh toan
    bgt $t7,99,printOp      # Neu Postfix hien tai > 99 --> 1 toan tu
    # Neu khong thi Postfix hien tai la mot so
    li $v0, 1
    add $a0,$t7,$zero
    syscall
    li $v0, 11
    li $a0, ' '
    syscall
    j printPost             # Loop
printOp:
    li $v0, 11
    addi $t7,$t7,-100       #Giai ma toan tu
    add $a0,$t7,$zero
    syscall
    li $v0, 11
    li $a0, ' '
    syscall
    j printPost             # Loop
finishPrint:
    li $v0, 11
    li $a0, '\n'
    syscall
    # Tinh toan
    li $t9,-4              # Dat top cua stack offset ve -4

```


la \$t3,stack # Load dia chi stack
li \$t6,-1 # Load Postfix offset hien tai ve -1

calPost:

addi \$t6,\$t6,1 # Tang Postfix offset hien tai
add \$t8,\$t2,\$t6 # Load dia chi cua Postfix hien tai
lbu \$t7,(\$t8) # Load gia tri cua Postfix hien tai
bgt \$t6,\$t5,printResult # Tinh toan moi postfix --> in
bgt \$t7,99,calculate # Neu Postfix hien tai > 99 --> 1 toan tu --> popout 2 so de tinh
Neu khong thi Postfix hien tai la mot so
addi \$t9,\$t9,4 # stack top offset hien tai
add \$t4,\$t3,\$t9 # Dia chi stack top hien tai
sw \$t7,(\$t4) #day so vao stack
j calPost # Loop

calculate:

Pop 1 so
add \$t4,\$t3,\$t9
lw \$s3,(\$t4)
Pop so ke tiep
addi \$t9,\$t9,-4
add \$t4,\$t3,\$t9
lw \$s2,(\$t4)
Giai ma toan tu
beq \$t7,143,plus
beq \$t7,145,minus
beq \$t7,142,multiply
beq \$t7,147,divide
beq \$t7,137,mod

plus:

```

add $s1,$s2,$s3
sw $s1,($t4)
sub $s2,$s2,$s2 # Reset s2 s3
sub $s3,$s3,$s3
j calPost
minus:
sub $s1,$s2,$s3
sw $s1,($t4)
sub $s2,$s2,$s2 # Reset s2 s3
sub $s3,$s3,$s3
j calPost
multiply:
mul $s1,$s2,$s3
sw $s1,($t4)
sub $s2,$s2,$s2 # Reset s2 s3
sub $s3,$s3,$s3
j calPost
divide:
div $s1,$s2,$s3
sw $s1,($t4)
sub $s2,$s2,$s2 # Reset s2 s3
sub $s3,$s3,$s3
j calPost
mod:
div $s1,$s2,$s3
mfhi $s1
sw $s1,($t4)
sub $s2,$s2,$s2 # Reset s2 s3
sub $s3,$s3,$s3

```

```

j calPost
printResult:
li $v0, 4
la $a0, prompt_result
syscall
li $v0, 1
lw $a0,($t4)
syscall
li $v0, 11
li $a0, '\n'
syscall
ask:                # Hoi nguoi dung co tiep tuc khong
li $v0, 50
la $a0, endMsg
syscall
beq $a0,0,start
beq $a0,2,ask
# Ket thuc chuong trinh
end:
li $v0, 55
la $a0, byeMsg
li $a1, 1
syscall
li $v0, 10
syscall

# Chuong trinh con
EOF:
beq $s7,2,wrongInput                # Ket thuc voi mot toan tu hoac dau mo ngoac

```

```

beq $s7,3,wrongInput
beq $t5,-1,wrongInput          # Khong nhap vao anything
j popAll
digit1:
beq $t4,'0',store1Digit
beq $t4,'1',store1Digit
beq $t4,'2',store1Digit
beq $t4,'3',store1Digit
beq $t4,'4',store1Digit
beq $t4,'5',store1Digit
beq $t4,'6',store1Digit
beq $t4,'7',store1Digit
beq $t4,'8',store1Digit
beq $t4,'9',store1Digit
j continueScan

digit2:
beq $t4,'0',store2Digit
beq $t4,'1',store2Digit
beq $t4,'2',store2Digit
beq $t4,'3',store2Digit
beq $t4,'4',store2Digit
beq $t4,'5',store2Digit
beq $t4,'6',store2Digit
beq $t4,'7',store2Digit
beq $t4,'8',store2Digit
beq $t4,'9',store2Digit
# Neu khong nhan chu so thu 2
jal numberToPost

```

j continueScan

digit3:

Neu scan chu so thu 3 --> error

beq \$t4,'0',wrongInput

beq \$t4,'1',wrongInput

beq \$t4,'2',wrongInput

beq \$t4,'3',wrongInput

beq \$t4,'4',wrongInput

beq \$t4,'5',wrongInput

beq \$t4,'6',wrongInput

beq \$t4,'7',wrongInput

beq \$t4,'8',wrongInput

beq \$t4,'9',wrongInput

Neu khong nhan chu so thu 3

jal numberToPost

j continueScan

plusMinus: # Input la + -

beq \$s7,2,wrongInput # Nhan vao 1 toan tu ngay sau 1 toan tu hoac nhan vao dau mo ngoac

beq \$s7,3,wrongInput

beq \$s7,0,wrongInput # Nhan vao toan tu truoc bat ky so nao

li \$s7,2 # Chuyen trang thai input sang 2

continuePlusMinus:

beq \$t6,-1,inputToOp # Khong co gi trong Operator stack --> day vao

add \$t8,\$t6,\$t3 # Load dia chi cua top Operator

lb \$t7,(\$t8) # Load gia tri byte cua top Operator

beq \$t7,'(',inputToOp # Neu top la (--> day vao

beq \$t7,'+',equalPrecedence # Neu top la + -

beq \$t7,'-',equalPrecedence

beq \$t7,'*',lowerPrecedence # Neu top la * / %

```

beq $t7, '/', lowerPrecedence
beq $t7, '%', lowerPrecedence
multiplyDivide:                # Input la * /
beq $s7, 2, wrongInput         # Nhan vao 1 toan tu ngay sau 1 toan tu hoac nhan vao dau mo ngoac
beq $s7, 3, wrongInput
beq $s7, 0, wrongInput         # Nhan vao toan tu truoc bat ky so nao
li $s7, 2                      # Chuyen trang thai input sang 2
beq $t6, -1, inputToOp         # Khong co gi trong Operator stack --> day vao
add $t8, $t6, $t3              # Load dia chi cua top Operator
lb $t7, ($t8)                  # Load gia tri byte cua top Operator
beq $t7, '(', inputToOp        # Neu top la ( --> day vao
beq $t7, '+', inputToOp        # Neu top la + --> day vao
beq $t7, '-', inputToOp
beq $t7, '*', equalPrecedence  # Neu top la * / %
beq $t7, '/', equalPrecedence
beq $t7, '%', equalPrecedence
openBracket:                   # Input la (
beq $s7, 1, wrongInput         # Nhan vao 1 dau mo ngoac sau 1 so hoac nhan vao dau dong ngoac
beq $s7, 4, wrongInput
li $s7, 3                      # Chuyen trang thai input sang 3
j inputToOp
closeBracket:                  # Input la )
beq $s7, 2, wrongInput         # Nhan vao 1 dau dong ngoac sau 1 toan tu hoac nhan vao 1 toan tu
beq $s7, 3, wrongInput
li $s7, 4                      # Chuyen trang thai input sang 4
add $t8, $t6, $t3              # Load dia chi cua top Operator
lb $t7, ($t8)                  # Load gia tri byte cua top Operator
beq $t7, '(', wrongInput       # Input chua ( ) ma khong co gi ben trong --> error
continueCloseBracket:

```

beq \$t6,-1,wrongInput	# Không thể tìm 1 dấu ngoặc --> error
add \$t8,\$t6,\$t3	# Load địa chỉ của top Operator
lb \$t7,(\$t8)	# Load giá trị byte của top Operator
beq \$t7,'(',matchBracket	# Tìm dấu ngoặc tương ứng
jal opToPostfix	# Pop top của Operator sang Postfix
j continueCloseBracket	# Sau đó loop lại cho đến khi tìm được dấu ngoặc tương ứng hoặc error

equalPrecedence:	# Nhập vào + - và top là + - nhập vào * / % và top là * / %
------------------	--

jal opToPostfix	# Pop top của Operator sang Postfix
-----------------	-------------------------------------

j inputToOp	# Đặt toán tử mới vào
-------------	-----------------------

lowerPrecedence:	# Nhập vào + - và top là * / %
------------------	--------------------------------

jal opToPostfix	# Pop top của Operator sang Postfix
-----------------	-------------------------------------

j continuePlusMinus	# Loop again
---------------------	--------------

inputToOp:	# Đặt input sang Operator
------------	---------------------------

add \$t6,\$t6,1	# Tăng top của Operator offset
-----------------	--------------------------------

add \$t8,\$t6,\$t3	# Load địa chỉ của top Operator
--------------------	---------------------------------

sb \$t4,(\$t8)	# Store input vào Operator
----------------	----------------------------

j scanInfix	
-------------	--

opToPostfix:	# Pop top của Operator rồi đặt vào Postfix
--------------	--

addi \$t5,\$t5,1	# Tăng top của Postfix offset
------------------	-------------------------------

add \$t8,\$t5,\$t2	# Load địa chỉ của top Postfix
--------------------	--------------------------------

addi \$t7,\$t7,100	# Giá trị toán tử + 100
--------------------	-------------------------

sb \$t7,(\$t8)	# Store toán tử vào Postfix
----------------	-----------------------------

addi \$t6,\$t6,-1	# Giảm top của Operator offset
-------------------	--------------------------------

jr \$ra	
---------	--

matchBracket:	# Loại bỏ 1 cặp dấu ngoặc tương ứng
---------------	-------------------------------------

addi \$t6,\$t6,-1	# Giảm top của Operator offset
-------------------	--------------------------------

j scanInfix	
-------------	--

popAll:	# Pop mọi Operator sang Postfix
---------	---------------------------------

```

jal numberToPost
beq $t6,-1,finishScan      # Operator trong --> finish
add $t8,$t6,$t3            # Load dia chi top Operator
lb $t7,($t8)               # Load gia tri byte cua top Operator
beq $t7,(' ',wrongInput    # Dau ngoac khong tuong ung --> error
beq $t7,')',wrongInput
jal opToPostfix
j popAll                   # Loop cho den khi Operator rong
store1Digit:
beq $s7,4,wrongInput      # Nhan vao so sau )
addi $s4,$t4,-48           # Store chu so thu nhat nhu 1 so
add $t9,$zero,1           # Chuyen trang thai sang 1 chu so
li $s7,1
j scanInfix
store2Digit:
beq $s7,4,wrongInput      # Nhan vao so sau )
addi $s5,$t4,-48           # Store chu so thu 2 nhu 1 so
mul $s4,$s4,10
add $s4,$s4,$s5            # Stored number = first digit * 10 + second digit
add $t9,$zero,2           # Chuyen trang thai sang 2 chu so
li $s7,1
j scanInfix
numberToPost:
beq $t9,0,endnumberToPost
addi $t5,$t5,1
add $t8,$t5,$t2
sb $s4,($t8)              # Store so trong Postfix
add $t9,$zero,$zero       # Chuyen trang thai sang 0 chu so
endnumberToPost:

```


jr \$ra