TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Final Project
Computer Architecture

GVHD: ThS. Lê Bá Vui

Nhóm: 19

Thành Viên: Mã Chung Thành - 20205026

Mã Lớp: 130939

Bài 7: chương trình kiểm tra cú pháp lệnh MIPS

Đề bài: Trình biên dịch của bộ xử lý MIPS sẽ tiến hành kiểm tra cú pháp các lệnh hợp ngữ trong mã nguồn, xem có phù hợp về cú pháp hay không, rồi mới tiến hành dịch các lệnh ra mã máy. Hãy viết một chương trình kiểm tra cú pháp của 1 lệnh hợp ngữ MIPS bất kì (không làm với giả lệnh) như sau:

-Nhập vào từ bàn phím một dòng lệnh hợp ngữ. Ví dụ: beq s1,31,t4

-Kiểm tra xem mã opcode có đúng hay không? Trong ví dụtrên, opcode là beq là hợp lệ thì hiển thị thông báo “opcode: beq, hợp lệ”.

-Kiểm tra xem tên các toán hạng phía sau có hợp lệ hay không? Trong ví dụ trên, toán hạng s1 là hợp lệ, 31 là không hợp lệ, t4 thì khỏi phải kiểm tra nữa vì toán hạng trước đã bị sai rồi.Gợi ý: nên xây dựng một cấu trúc chứa khuôn dạng của từng lệnh với tên lệnh, kiểu của toán hạng 1, toán hạng 2, toán hạng 3.

Source code:

```
.data
Message1: .asciiz "Nhap dong lenh can check: "
Message2: .asciiz "\nOpcode: "
Message3: .asciiz ", hop le!"
```

```
Message4: .asciiz " khong hop le!"

Message5: .asciiz " \nCau lenh dung!\n----------\n"

Message6: .asciiz " \nCau lenh sai!\n----------\n"

Message7: .asciiz " \n"

Message8: .asciiz "Thanh ghi "

Message9: .asciiz "So "

Message10: .asciiz "Nhan "

Message11: .asciiz "Ban muon kiem tra tiep khong?"

string: .space 100

#Luu cac opcode can check vao mang

Opcode_R_Check: .asciiz
"/add/sub/addu/subu/and/or/slt/sltu/nor/srav/srlv/movn/movz/mul/ "

Opcode_R_Check_1: .asciiz "/beq/bne/ "

Opcode_R_Check_2: .asciiz
"/div/divu/mfc0/mult/multu/clo/clz/move/negu/not/madd/maddu/msub/msubu/ "

Opcode_I_Check: .asciiz "/addi/addiu/andi/ori/slti/sltiu/sll/srl/sra/ "

Opcode_I_Check_1: .asciiz "/li/lui/ "

Opcode_J_Check: .asciiz "/j/jal/ "

Opcode_J_Check_1: .asciiz "/jr/mfhi/mthi/mflo/mtlo/ "

Opcode_L_Check: .asciiz "/lb/lbu/lhu/ll/lw/sb/sc/sh/sw/lwc1/ldc1/swc1/sdc1/ "

Opcode_L_Check_1: .asciiz "/la/ "

Special_command: .asciiz "/syscall/nop/ "

Register_Check: .asciiz
```

```
"/$zero/$at/$v0/$v1/$a0/$a1/$a2/$a3/$t0/$t1/$t2/$t3/$t4/$t5/$t6/$t7/$s0/$s1/$s2/$s3/$s
4/$s5/$s6/$s7/$t8/$t9/$k0/$k1/$gp/$sp/$sp/$fp/$ra/$0/$1/$2/$3/$4/$5/$6/$7/$8/$9/$10/
$11/$12/$13/$14/$15/$16/$17/$18/$19/$20/$21/$22/$23/$24/$25/$26/$27/$28/$29/$30/
$31/ " chain_check: .word  #Chua xau ki tu

đang xet

.text start:

        la      $s2, chain_check #Dia chi chua chain_check

        li      $s6, 32                 #s6=space

li      $s7, 47                 #s7 = '/'

#Nhap dong lenh can check

 li  $v0, 54  la  $a0, Message1

la   $a1,  string   la   $a2,  100

syscall  la  $s1, string

#-------------------------------------------------

#main
        jal     Split_opcode

        jal     Check_opcode

        beq     $s4, $zero, False_opcode #Opcode false

        addi    $t0, $zero, 5                   #Syscall, nop->Right code

        beq     $s4, $t0, Right_code

        addi    $t5, $zero, 1
        beq     $s4, $t5, R_Check_Register_and_Number

        addi    $t5, $zero, 2

        beq     $s4, $t5, R_1_Check_Register_and_Number
```

```
        addi    $t5, $zero, 3
        beq     $s4, $t5, I_Check_Register_and_Number
        addi    $t5, $zero, 4
        beq     $s4, $t5, J_Check_Register_and_Number
        addi    $t5, $zero, 6
        beq     $s4, $t5, R_2_Check_Register_and_Number
        addi    $t5, $zero, 7
        beq     $s4, $t5, I_1_Check_Register_and_Number
        addi    $t5, $zero, 8
        beq     $s4, $t5, J_1_Check_Register_and_Number
        addi    $t5, $zero, 9
        beq     $s4, $t5, L_Check_Register_and_Number
        addi    $t5, $zero, 10
        beq     $s4, $t5, L_1_Check_Register_and_Number
        j       End_main
#-------------------------------------------------

#Tach ma opcode Split_opcode:
        li      $s5, 0          #Vi tri load ban dau cua lenh nap vao
        li      $s0, 0          #Vi tri phan tu cuoi cua mang chain_check
        li      $t1, 0 #i=0
Loop1:
        add     $a2, $s1, $t1 #a2 = Dia chi cua ky tu dang load
        add     $a3, $s2, $s0           #a3 = Dia chi dang nap vao hang doi
```

```
        lb      $t0, 0($a2)
        beq     $t0, $zero, EndLoop         #Gap null => ket thuc vong lap 1
        beq     $t0, $s6, Loop1_them
        sb      $t0, 0($a3)             #Nap ky tu vao hang doi
        addi    $s0, $s0, 1              #Dich chuyen vi tri cuoi cua hang doi sang phai
        addi    $t1, $t1, 1
        addi    $s5, $s5, 1
Loop2:
        add     $a2, $s1, $t1 #a2 = Dia chi cua ky tu dang load
        add     $a3, $s2, $s0           #a3 = Dia chi dang nap vao hang doi
        lb      $t0, 0($a2)
        beq     $t0, $zero, EndLoop         #Gap null => ket thuc vong lap 1
        beq     $t0, $s6, EndLoop #Gap space => ket thuc vong lap 1
        li      $t5, 10                     #t5=newline
        beq     $t0, $t5, EndLoop #Gap newline => ket thuc vong lap 1
        sb      $t0, 0($a3)             #Nap ky tu vao hang doi
        addi    $s0, $s0, 1              #Dich chuyen vi tri cuoi cua hang doi sang phai
        addi    $t1, $t1, 1
        addi    $s5, $s5, 1
        j       Loop2
EndLoop:
        #Chen ky tu NULL cho hang doi
sb      $zero, 0($a3)
```

```
 #add $s5, $s0, $zero  #Luu vi tri ki tu dang doc vao s5  addi $s0, $s0, -1

        jr      $ra

#-------------------------------------------------

#Tach ma thanh ghi va so

Split_Register_and_Number:

        li      $s0, 0          #Vi tri phan tu cuoi cua mang chain_check
add     $t1, $s5, $zero         #i=vi tri dang doc trong cau lenh=s5
Loop1_Split:
        add     $a2, $s1, $t1 #a2 = Dia chi cua ky tu dang load

        add     $a3, $s2, $s0           #a3 = Dia chi dang nap vao hang doi

        lb      $t0, 0($a2)     #t0 = Ky tu dang Load

        add     $t9, $zero, $t0         #t9 = Ky tu cuoi cung duoc load

        beq     $t0, $zero, EndLoop_Split#Check_Reg_and_Num       #Gap null => ket
thuc vong lap 1

        beq     $t0, $s6, Loop1_Split_them          #Gap Space -> Chay qua Space

li      $t5, 44                 #t5=44~'dau phay,'

        beq     $t0, $t5, False_code

        sb      $t0, 0($a3)     #Nap ky tu vao hang doi     addi $s0, $s0, 1       #Dich

chuyen vi tri cuoi cua hang doi sang phai

          addi $t1, $t1, 1

Loop2_Split:

        add     $a2, $s1, $t1 #a2 = Dia chi cua ky tu dang load

add     $a3, $s2, $s0           #a3 = Dia chi dang nap vao hang doi      lb

$t0, 0($a2)
```

```
        add     $t9, $zero, $t0         #t9 = Ky tu cuoi cung duoc load   beq     $t0,
$zero, EndLoop_Split#Check_Reg_and_Num   #Gap null => ket thuc vong lap 1
        beq     $t0, $s6, Loop3_Split         #Gap space => Chay qua Space
        li      $t5, 10                       #t5=newline
        beq     $t0, $t5, EndLoop_Split      #Check_Reg_and_Num     #Gap newline =>
ket thuc vong lap 1
        li      $t5, 44                       #t5=44~'dau phay,'
        beq     $t0, $t5, EndLoop_Split      #Gap dau phay => ket thuc vong lap 1
        sb      $t0, 0($a3)             #Nap ky tu vao hang doi
        addi    $s0, $s0, 1              #Dich chuyen vi tri cuoi cua hang doi sang phai
        addi    $t1, $t1, 1
        j       Loop2_Split
Loop3_Split:
        add     $a2, $s1, $t1 #a2 = Dia chi cua ky tu dang load
add     $a3, $s2, $s0          #a3 = Dia chi dang nap vao hang doi      lb
$t0, 0($a2)    #t0 = Ky tu dang Load       add     $t9, $zero, $t0
#t9 = Ky tu cuoi cung duoc load   beq     $t0, $zero,
EndLoop_Split#Check_Reg_and_Num   #Gap null => ket thuc vong
lap 1
        beq     $t0, $s6, Loop3_Split_them          #Gap Space -> Chay qua Space
li      $t5, 44                 #t5=44~'dau phay,'
        beq     $t0, $t5, EndLoop_Split
        li      $t5, 10                 #t5=10~'New line'
        beq     $t0, $t5, EndLoop_Split
```

```
        j       False_code
EndLoop_Split:
        #Chen ky tu NULL cho hang doi
sb      $zero, 0($a3)
 addi $s5, $t1, 1 #Luu vi tri ki tu dang doc vao s5  addi $s0, $s0,
-1
        jr      $ra
#-------------------------------------------------
#Tach Sign ExtImm Split_Sign_ExtImm:
        li      $s0, 0          #Vi tri phan tu cuoi cua mang chain_check
add     $t1, $s5, $zero         #i=vi tri dang doc trong cau lenh=s5
Loop1_Sign:
        add     $a2, $s1, $t1 #a2 = Dia chi cua ky tu dang load
        add     $a3, $s2, $s0           #a3 = Dia chi dang nap vao hang doi      lb      $t0,
0($a2)          #t0 = Ky tu dang Load       add     $t9, $zero, $t0         #t9 = Ky tu cuoi
cung duoc load      beq     $t0, $zero, EndLoop_Sign_them_2#Check_Reg_and_Num
#Gap null =>
ket thuc vong lap 1
        li      $t5, 10                 #t5=10~'New line'
        beq     $t0, $t5, EndLoop_Sign_them_2
        beq     $t0, $s6, Loop1_Sign_them           #Gap Space -> Chay qua Space
        li      $t5, 44                 #t5=44~'dau phay,'
        beq     $t0, $t5, False_code
```

```
        sb      $t0, 0($a3)             #Nap ky tu vao hang doi

        li      $t5, 40                 #Thay dau ( thi ket thuc

        beq     $t0, $t5, EndLoop_Sign_them

        li      $t5, 41                 #Thay dau ) thi ket thuc

        beq     $t0, $t5, EndLoop_Sign_them_3

        addi    $s0, $s0, 1             #Dich chuyen vi tri cuoi cua hang doi sang phai

        addi    $t1, $t1, 1
Loop2_Sign:  add $a2, $s1, $t1 #a2 = Dia chi cua ky tu dang load  add

$a3, $s2, $s0 #a3 = Dia chi dang nap vao hang doi  lb $t0, 0($a2)

        add     $t9, $zero, $t0         #t9 = Ky tu cuoi cung duoc load
 beq $t0, $zero, EndLoop_Sign_them_2#Check_Reg_and_Num #Gap null => ket thuc
vong lap 1

        li      $t5, 10                 #t5=10~'New line'

        beq     $t0, $t5, EndLoop_Sign_them_2           beq     $t0, $s6,

EndLoop_Sign        #Gap space => Chay qua Space

        li      $t5, 10                 #t5=newline

        beq     $t0, $t5, EndLoop_Sign      #Check_Reg_and_Num      #Gap newline =>
ket thuc vong lap 1
        li      $t5, 44                 #t5=44~'dau phay,'

        beq     $t0, $t5, EndLoop_Sign      #Gap dau phay => ket thuc vong lap 1

        li      $t5, 40                 #Thay dau ( thi ket thuc

        beq     $t0, $t5, EndLoop_Sign_them_1

        li      $t5, 41                 #Thay dau ) thi ket thuc

        beq     $t0, $t5, EndLoop_Sign_them_1
```

```
        sb      $t0, 0($a3)             #Nap ky tu vao hang doi
        addi    $s0, $s0, 1             #Dich chuyen vi tri cuoi cua hang doi sang phai
        addi    $t1, $t1, 1
        j       Loop2_Sign
EndLoop_Sign:
        #Chen ky tu NULL cho hang
doi     sb      $zero, 0($a3)       addi
$s5, $t1, 0     addi $s0, $s0, -1

        jr      $ra

#-------------------------------------------------
#Check Opcode
Check_opcode:
 li $s4, 0 #s4 bieu thi cho khuon dang lenh: Saiopcode: 0, R: 1, R_1: 2, I: 3, J: 4, Dac biet: 5

        #Check_R
        la      $s3, Opcode_R_Check
        li      $t1, 0 #i=0
Loop1_R:
        add     $a3, $s3, $t1 #load byte cua opcode mau
        lb      $t3, 0($a3)
         addi $t1, $t1, 1
        bne     $t3, $s7, Loop1_R
        li      $t0, 0          #So ki tu cua opcode mau
Loop2_R:
```

```
        add     $a3, $s3, $t1 #load byte cua opcode mau
        lb      $t3, 0($a3)
        add     $a2, $s2, $t0 #Load byte cua opcode can check
        lb      $t2, 0($a2)
        beq     $t3, $s7, Check_R
        beq     $t3, $s6, End_Loop_R
        bne     $t2, $t3, Loop1_R_them     #Kiem tra xem opcode check va opcode mau
co giong nhau khong
            beq  $t2, $t3, Loop2_R_them
End_Loop_R:



        #Check_R_2          la      $s3,
Opcode_R_Check_2        li      $t1,
0 #i=0
Loop1_R_2:          add     $a3, $s3, $t1 #load byte
cua opcode mau
        lb      $t3, 0($a3)    addi $t1, $t1, 1        bne
$t3, $s7, Loop1_R_2         li      $t0, 0          #So
ki tu cua opcode mau
Loop2_R_2:
        add     $a3, $s3, $t1 #load byte cua opcode mau
        lb      $t3, 0($a3)
        add     $a2, $s2, $t0 #Load byte cua opcode can check
        lb      $t2, 0($a2)
        beq     $t3, $s7, Check_R_2
        beq     $t3, $s6, End_Loop_R_2
```

```
        bne     $t2, $t3, Loop1_R_2_them #Kiem tra xem opcode check va opcode
mau co giong nhau khong         beq     $t2, $t3, Loop2_R_2_them
End_Loop_R_2:




        #Check_I
        la      $s3, Opcode_I_Check

        li      $t1, 0 #i=0
Loop1_I:
        add     $a3, $s3, $t1 #load byte cua opcode mau

        lb      $t3, 0($a3)

        addi    $t1, $t1, 1

        bne     $t3, $s7, Loop1_I

        li      $t0, 0          #So ki tu cua opcode mau
Loop2_I:

        add     $a3, $s3, $t1 #load byte cua opcode mau

        lb      $t3, 0($a3)

        add     $a2, $s2, $t0 #Load byte cua opcode can check

        lb      $t2, 0($a2)

        beq     $t3, $s7, Check_I

        beq     $t3, $s6, End_Loop_I
 bne $t2, $t3, Loop1_I_them #Kiem tra xem opcode check va opcode mau co giong
nhau khong
          beq  $t2, $t3, Loop2_I_them
End_Loop_I:
```

```
        #Check_I_1
        la      $s3, Opcode_I_Check_1

        li      $t1, 0 #i=0

Loop1_I_1:

        add     $a3, $s3, $t1 #load byte cua opcode mau

        lb      $t3, 0($a3)

        addi    $t1, $t1, 1

        bne     $t3, $s7, Loop1_I_1

        li      $t0, 0          #So ki tu cua opcode mau

Loop2_I_1:

        add     $a3, $s3, $t1 #load byte cua opcode mau

        lb      $t3, 0($a3)

        add     $a2, $s2, $t0 #Load byte cua opcode can check

        lb      $t2, 0($a2)

        beq     $t3, $s7, Check_I_1

        beq     $t3, $s6, End_Loop_I_1

        bne     $t2, $t3, Loop1_I_1_them #Kiem tra xem opcode check va opcode mau
co giong nhau khong

        beq     $t2, $t3, Loop2_I_1_them

End_Loop_I_1:



        #Check_J
        la      $s3, Opcode_J_Check

        li      $t1, 0 #i=0
```

```
Loop1_J:
        add     $a3, $s3, $t1 #load byte cua opcode mau

        lb      $t3, 0($a3)

        addi    $t1, $t1, 1

        bne     $t3, $s7, Loop1_J
        li      $t0, 0          #So ki tu cua opcode mau

Loop2_J:

        add     $a3, $s3, $t1 #load byte cua opcode mau

        lb      $t3, 0($a3)

        add     $a2, $s2, $t0 #Load byte cua opcode can check

        lb      $t2, 0($a2)

        beq     $t3, $s7, Check_J

        beq     $t3, $s6, End_Loop_J

        bne     $t2, $t3, Loop1_J_them      #Kiem tra xem opcode check va opcode mau
co giong nhau khong
           beq  $t2, $t3, Loop2_J_them
End_Loop_J:



        #Check_J_1          la      $s3,

Opcode_J_Check_1        li      $t1,

0 #i=0

Loop1_J_1:  add     $a3, $s3, $t1 #load byte cua

opcode mau
```

```
        lb      $t3, 0($a3)   addi $t1, $t1, 1       bne
$t3, $s7, Loop1_J_1       li      $t0, 0          #So
ki tu cua opcode mau
Loop2_J_1:  add    $a3, $s3, $t1 #load byte cua
opcode mau
        lb      $t3, 0($a3)
        add     $a2, $s2, $t0 #Load byte cua opcode can check
 lb $t2, 0($a2)  beq $t3, $s7, Check_J_1  beq $t3, $s6, End_Loop_J_1  bne $t2, $t3,
Loop1_J_1_them #Kiem tra xem opcode check va opcode mau co giong nhau khong
beq $t2, $t3, Loop2_J_1_them
End_Loop_J_1:


        #Check Special Command
        la      $s3, Special_command
        li      $t1, 0 #i=0
Loop1_Sc:
        add     $a3, $s3, $t1 #load byte cua opcode mau
        lb      $t3, 0($a3)
        addi    $t1, $t1, 1
        bne     $t3, $s7, Loop1_Sc
        li      $t0, 0          #So ki tu cua opcode mau
Loop2_Sc:
        add     $a3, $s3, $t1 #load byte cua opcode mau
        lb      $t3, 0($a3)
```

```
        add     $a2, $s2, $t0 #Load byte cua opcode can check
        lb      $t2, 0($a2)
        beq     $t3, $s7, Check_Sc
        beq     $t3, $s6, End_Loop_Sc
        bne     $t2, $t3, Loop1_Sc_them #Kiem tra xem opcode check va opcode mau
co giong nhau khong
        beq     $t2, $t3, Loop2_Sc_them
End_Loop_Sc:


        #Check_L    la      $s3,
Opcode_L_Check
        li      $t1, 0 #i=0
Loop1_L:
        add     $a3, $s3, $t1 #load byte cua opcode mau
        lb      $t3, 0($a3)
         addi $t1, $t1, 1
        bne     $t3, $s7, Loop1_L
        li      $t0, 0          #So ki tu cua opcode mau
Loop2_L:
        add     $a3, $s3, $t1 #load byte cua opcode mau
        lb      $t3, 0($a3)
        add     $a2, $s2, $t0 #Load byte cua opcode can check
        lb      $t2, 0($a2)
        beq     $t3, $s7, Check_L
        beq     $t3, $s6, End_Loop_L
        bne     $t2, $t3, Loop1_L_them     #Kiem tra xem opcode check va opcode mau
co giong nhau khong
           beq  $t2, $t3, Loop2_L_them
```

```
End_Loop_L:
        #Check_L_1          la      $s3,
Opcode_L_Check_1        li      $t1,
0 #i=0
Loop1_L_1:
        add     $a3, $s3, $t1 #load byte cua opcode mau
        lb      $t3, 0($a3)
        addi $t1, $t1, 1        bne     $t3, $s7,
Loop1_L_1   li      $t0, 0          #So ki tu cua
opcode mau
Loop2_L_1:  add     $a3, $s3, $t1 #load byte cua
opcode mau
        lb      $t3, 0($a3)
        add     $a2, $s2, $t0 #Load byte cua opcode can check
 lb $t2, 0($a2)  beq $t3, $s7, Check_L_1  beq $t3, $s6, End_Loop_L_1  bne $t2, $t3,
Loop1_L_1_them #Kiem tra xem opcode check va opcode mau co giong nhau khong
beq $t2, $t3, Loop2_L_1_them
End_Loop_L_1:


#Check_R_1              la      $s3,
Opcode_R_Check_1  li $t1, 0 #i=0
Loop1_R_1:          add     $a3, $s3, $t1 #load byte
cua opcode mau
```

```
        lb      $t3, 0($a3)
addi $t1, $t1, 1
        bne     $t3,        $s7,
                Loop1_R_1
        li
        $t0, 0
        #So ki
tu cua opcode
mau
Loop2_R_1:
        add     $a3, $s3, $t1 #load
                byte cua opcode
                mau

        lb      $t3, 0($a3)

                $a2, $s2, $t0 #Load
        add     byte cua opcode can
                check

        lb      $t2, 0($a2)

        beq     $t3, $s7,
                Check_R_1
        beq     $t3, $s6,
                End_Loop_R_1
                $t2,            $t3,
                Loop1_R_1_them
                #Kiem    tra    xem
                opcode  check  va
        bne     opcode mau
co giong nhau khong     beq     $t2,

$t3, Loop2_R_1_them

End_Loop_R_1:

        jr      $ra

#-------------------------------------------------

#Check cac thanh ghi va so

R_Check_Register_and_Number:
        jal     Right_opcode
```

```
        jal     Split_Register_and_Number

        jal     Check_Register

        #jal    Check_Number

        jal     Split_Register_and_Number

        jal     Check_Register
        jal     Split_Register_and_Number
        jal     Check_Register

        addi    $t5, $zero, 10

        beq     $t9, $t5, Right_code

        addi    $t5, $zero, 0

        beq     $t9, $t5, Right_code

        j       False_code
R_1_Check_Register_and_Number:
        jal     Right_opcode

        jal     Split_Register_and_Number

        jal     Check_Register

        jal     Split_Register_and_Number

        jal     Check_Register

        jal     Split_Register_and_Number

        addi    $t5, $zero, 10

        beq     $t9, $t5, R_1_Check_Label

        addi    $t5, $zero, 0

        beq     $t9, $t5, R_1_Check_Label

        j       False_code
```

```
        R_1_Check_Label:

jal     Check_Label

R_2_Check_Register_and_Number:

        jal     Right_opcode
        jal     Split_Register_and_Number
        jal     Check_Register

        jal     Split_Register_and_Number

        jal     Check_Register

        addi    $t5, $zero, 10

        beq     $t9, $t5, Right_code

        addi    $t5, $zero, 0

        beq     $t9, $t5, Right_code

        j       False_code
I_Check_Register_and_Number:
        jal     Right_opcode

        jal     Split_Register_and_Number

        jal     Check_Register

        #jal    Check_Number

        jal     Split_Register_and_Number

        jal     Check_Register

        jal     Split_Register_and_Number

        jal     Check_Number

        addi    $t5, $zero, 10

        beq     $t9, $t5, Right_code
```

```
        addi    $t5, $zero, 0
        beq     $t9, $t5, Right_code
        j       False_code
I_1_Check_Register_and_Number:
        jal     Right_opcode
        jal     Split_Register_and_Number
        jal     Check_Register
        #jal    Check_Number
        jal     Split_Register_and_Number
        jal     Check_Number
        addi    $t5, $zero, 10
        beq     $t9, $t5, Right_code
        addi    $t5, $zero, 0
        beq     $t9, $t5, Right_code
        j       False_code
J_Check_Register_and_Number:
        jal     Right_opcode
        jal     Split_Register_and_Number
        addi    $t5, $zero, 10
        beq     $t9, $t5, J_Check_Label
        addi    $t5, $zero, 0
        beq     $t9, $t5, J_Check_Label
        j       False_code
```

```
        J_Check_Label:

jal     Check_Label

J_1_Check_Register_and_Number:
      jal   Right_opcode
        jal  Split_Register_and_Number

        jal  Check_Register

       addi$t5, $zero, 10

        beq$t9, $t5, Right_code

       addi$t5, $zero, 0

        beq$t9, $t5, Right_code

        j    False_code
L_Check_Register_and_Number:

        jal     Right_opcode      jal

Split_Register_and_Number     jal

Check_Register     jal

Check_Sign_ExtImm

L_1_Check_Register_and_Number:
        jal     Right_opcode

        jal     Split_Register_and_Number

        jal     Check_Register

        jal     Split_Register_and_Number

        addi    $t5, $zero, 10

        beq     $t9, $t5, L_1_Check_Label

        addi    $t5, $zero, 0
```

```
        beq     $t9, $t5, L_1_Check_Label

        j       False_code
        L_1_Check_Label:
        jal     Check_Label

#-------------------------------------------------

Loop1_them:         addi

$t1, $t1, 1     addi  $s5,

$s5, 1

        j       Loop1

Loop1_Split_them:

addi $t1, $t1, 1

        j       Loop1_Split

Loop2_Split_them:

addi $t1, $t1, 1

        j       Loop2_Split

Loop3_Split_them:

addi $t1, $t1, 1

        j       Loop3_Split

Loop1_Sign_them:  addi

$s5, $s5, 1    addi $t1,

$t1, 1

        j       Loop1_Sign

Loop2_Sign_them:
```

```
        addi    $t1, $t1, 1

        j       Loop2_Sign
EndLoop_Sign_them:

addi $a3, $a3, 1      sb

$zero, 0($a3)         addi

$s5, $s5, 1

        jr      $ra

        #addi $s0, $s0, -1

        #addi $t1, $t1, 1
 #add   $a3,   $s2,   $s0     #j              #Cap nhat moi dia chi dang load cua hang doi

EndLoop_Sign

EndLoop_Sign_them_1:
                                              #Cap nhat moi dia chi dang load cua hang doi
 add   $a3,   $s2,   $s0      j

EndLoop_Sign

EndLoop_Sign_them_2:
        add     $s0, $s0, 1   j
        EndLoop_Sign

                                              #load cac ki tu sau dau ) de kiem tra dung sai

EndLoop_Sign_them_3:
        addi $a2, $a2, 1
        lb      $t9, 0($a2)


        li      $t5, 0              #t5 = NULL
```

```
        beq     $t9, $t5, Right_code

        li      $t5, 10                         #t5 = new line

        beq     $t9, $t5, Right_code
        li      $t5, 32                         #t5 = space
        beq     $t9, $t5, EndLoop_Sign_them_3

        j       False_code

Loop_Number_them:

addi $t1, $t1, 1

        j       Loop_Number

Loop_Number_them_1:

addi $t1, $t1, 1

        j       Loop_Number_1

Check_Mark_them:

addi $t1, $t1, 1

        j       Check_Mark_done


#Check thanh ghi R

Check_R:

 addi $t0, $t0, -1  beq $s0,

$t0, R_True

        j       Loop1_R

Loop1_R_them:

addi $t1, $t1, 1
```

```
        j       Loop1_R

Loop2_R_them:
addi $t1, $t1, 1
        addi    $t0, $t0, 1
        j       Loop2_R

R_True:

 li $s4, 1  jr $ra

#Check thanh ghi R_2

Check_R_2:          addi $t0,

$t0, -1         beq     $s0, $t0,

R_2_True

        j       Loop1_R_2

Loop1_R_2_them:

addi $t1, $t1, 1

        j       Loop1_R_2

Loop2_R_2_them:

addi $t1, $t1, 1

addi $t0, $t0, 1

        j       Loop2_R_2

R_2_True:

 li $s4, 6  jr $ra

#Check thanh ghi I Check_I:
```

```
        addi $t0, $t0, -1
beq     $s0, $t0, I_True
        j       Loop1_I
Loop1_I_them:
addi $t1, $t1, 1
        j       Loop1_I
Loop2_I_them:
addi $t1, $t1, 1
addi $t0, $t0, 1
        j       Loop2_I
I_True:
 li $s4, 3  jr $ra
#Check thanh ghi I_1
Check_I_1:   addi $t0, $t0, -1
beq     $s0, $t0, I_1_True
        j       Loop1_I_1
Loop1_I_1_them:
addi $t1, $t1, 1
        j       Loop1_I_1
Loop2_I_1_them:
        addi $t1, $t1, 1
        addi $t0, $t0, 1
```

```
        j       Loop2_I_1
I_1_True:
 li $s4, 7  jr $ra

#Check thanh ghi J Check_J:
addi $t0, $t0, -1       beq
$s0, $t0, J_True
        j       Loop1_J
Loop1_J_them:
addi $t1, $t1, 1
        j       Loop1_J
Loop2_J_them:
addi $t1, $t1, 1
addi $t0, $t0, 1
        j       Loop2_J
J_True:
 li $s4, 4  jr $ra

#Check thanh ghi J_1 Check_J_1:
 addi $t0, $t0, -1  beq $s0, $t0,
J_1_True
        j       Loop1_J_1
Loop1_J_1_them:
addi $t1, $t1, 1
```

```
        j       Loop1_J_1
Loop2_J_1_them:
addi $t1, $t1, 1
addi $t0, $t0, 1
        j       Loop2_J_1
J_1_True:
 li $s4, 8  jr $ra
#Check thanh ghi Sc - Special Command
Check_Sc:
 addi $t0, $t0, -1  beq $s0, $t0,
Sc_True
        j       Loop1_Sc
Loop1_Sc_them:
addi $t1, $t1, 1       j
Loop1_Sc

Loop2_Sc_them:
addi $t1, $t1, 1
addi $t0, $t0, 1
        j       Loop2_Sc
Sc_True:
 li $s4, 5  jr $ra
```

```
#Check thanh ghi R_1
Check_R_1:          addi $t0,
$t0, -1         beq     $s0, $t0,
R_1_True
        j       Loop1_R_1
Loop1_R_1_them:
addi $t1, $t1, 1
        j       Loop1_R_1
Loop2_R_1_them:
addi $t1, $t1, 1
addi $t0, $t0, 1
        j       Loop2_R_1
R_1_True:
        li      $s4, 2
        jr      $ra
#Check thanh ghi L Check_L:
        addi $t0, $t0, -1
beq     $s0, $t0, L_True
        j       Loop1_L
Loop1_L_them:
addi $t1, $t1, 1
        j       Loop1_L
```

```
Loop2_L_them:
addi $t1, $t1, 1
addi $t0, $t0, 1
        j       Loop2_L
L_True:
 li $s4, 9  jr $ra
#Check thanh ghi L_1
Check_L_1:  addi $t0, $t0, -1
beq     $s0, $t0, L_1_True
        j       Loop1_L_1
Loop1_L_1_them:
          addi $t1, $t1, 1
        j       Loop1_L_1
Loop2_L_1_them:
addi $t1, $t1, 1
addi $t0, $t0, 1
        j       Loop2_L_1
L_1_True:
 li $s4, 10  jr $ra
#-------------------------------------------------
#Check Register
Check_Register:
```

```
        la      $s3, Register_Check

li      $t1, 0 #i=0

Loop1_Reg:          add     $a3, $s3, $t1 #load byte cua

thanh ghi mau

        lb      $t3, 0($a3)   addi $t1, $t1, 1      bne
$t3, $s7, Loop1_Reg       li      $t0, 0          #So ki
tu cua thanh ghi mau Loop2_Reg:
        add   $a3, $s3, $t1 #load byte cua
              thanh ghi mau
        lb    $t3, 0($a3)

        add   $a2, $s2, $t0 #Load byte cua thanh ghi can
              check

        lb    $t2, 0($a2)

        beq   $t3, $s7, Check_Reg

        beq   $t3, $s6, False_code

              $t2, $t3, Loop1_Reg_them #Kiem tra xem
        bne   thanh ghi check va thanh ghi
mau co giong nhau khong  beq $t2, $t3,

Loop2_Reg_them

End_Loop_Reg:



#-------------------------------------------------

Check_Reg:

 addi $t0, $t0, -1  beq $s0, $t0,

Reg_True

        j       Loop1_Reg
```

```
Loop1_Reg_them:
addi $t1, $t1, 1
        j       Loop1_Reg
Loop2_Reg_them:
addi $t1, $t1, 1
addi $t0, $t0, 1
        j       Loop2_Reg
Reg_True:
        add  $t8, $zero, $ra
jal     Right_Register
        jr      $t8


#-------------------------------------------------
#Check Number
Check_Number:
        li      $t1, 0          #i = 0
j       Check_Mark
Check_Mark_done:
        add     $a2, $s2, $t1 #Kiem tra so dau tien
        lb      $t2, 0($a2)
        li      $t5, 10                 #t5 = newline
        beq     $t2, $t5, False_code
        beq     $t2, $zero, False_code
        li      $t5, 48                 #t5 = zero
```

```
        bne     $t2, $t5, Loop_Number_1
        slti    $t4, $t2, 48
        bne     $t4, $zero, False_code
        slti    $t4, $t2, 58
        beq     $t4, $zero, False_code
        addi    $t1, $t1, 1     #Kiem tra so thu hai(co the la x trong so hexa)
        add     $a2, $s2, $t1
        lb      $t2, 0($a2)
        beq     $t2, $zero, Right_Number
        li      $t5, 120
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 88
        beq     $t2, $t5, Loop_Number_them
        slti    $t4, $t2, 48
        bne     $t4, $zero, False_code
        slti    $t4, $t2, 58
        beq     $t4, $zero, False_code
Loop_Number:
        add     $a2, $s2, $t1
        lb      $t2, 0($a2)
        beq     $t2, $zero, Right_Number
        li      $t5, 48
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 49
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 50
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 51
        beq     $t2, $t5, Loop_Number_them
```

```
        li      $t5, 52
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 53
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 54
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 55
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 56
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 57
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 65
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 66
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 67
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 68
        beq     $t2, $t5, Loop_Number_them
        li      $t5, 69
        beq     $t2, $t5, Loop_Number_them

    li      $t5, 70
      beq$t2,                   $t5,
          Loop_Number_them
        li  $t5, 97
      beq$t2,                   $t5,
          Loop_Number_them
```

```
        li   $t5, 98
        beq  $t2, $t5, Loop_Number_them
        li   $t5, 99
        beq  $t2, $t5, Loop_Number_them
        li   $t5, 100
        beq  $t2, $t5, Loop_Number_them
        li   $t5, 101
        beq  $t2, $t5, Loop_Number_them
        li   $t5, 102
        beq  $t2, $t5, Loop_Number_them
        j    False_code
Loop_Number_1:
        add  $a2, $s2, $t1
        lb   $t2, 0($a2)
        beq  $t2, $zero, Right_Number
        li   $t5, 48
        beq  $t2, $t5, Loop_Number_them_1
        li   $t5, 49
        beq  $t2, $t5, Loop_Number_them_1
        li   $t5, 50
        beq  $t2, $t5, Loop_Number_them_1
        li   $t5, 51
        beq  $t2, $t5, Loop_Number_them_1
        li   $t5, 52
```

```
        beq     $t2, $t5, Loop_Number_them_1
        li      $t5, 53
        beq     $t2, $t5, Loop_Number_them_1
        li      $t5, 54
        beq     $t2, $t5, Loop_Number_them_1
        li      $t5, 55
        beq     $t2, $t5, Loop_Number_them_1
        li      $t5, 56
        beq     $t2, $t5, Loop_Number_them_1
        li      $t5, 57
        j       False_code
#-------------------------------------------------

Right_Number:
        add     $t8, $zero, $ra
jal     Print_Right_Number
        jr      $t8
#-------------------------------------------------

Check_Mark: #Ham kiem tra dau cua imm
        add     $a2, $s2, $t1 #Kiem tra xem ki tu dau tien cua Imm co phai dau + hay -
khong?

        lb      $t2, 0($a2)
        li      $t5, 43                 #t5 =43 ~ '+'
        beq     $t2, $t5, Check_Mark_them
```

```
        li      $t5, 45                 #t5 =45 ~ '-'

        beq     $t2, $t5, Check_Mark_them

        j       Check_Mark_done
#-------------------------------------------------

#Check Sign_ExtImm Check_Sign_ExtImm:
        add     $t8, $zero, $ra         #Luu dia chi tro ve chuong trinh vao -> t8

        jal     Split_Sign_ExtImm

        jal     Check_Number

        jal     Split_Sign_ExtImm

        jal     Check_Parentheses_1

        jal     Split_Sign_ExtImm

        jal     Check_Register

        jal     Split_Sign_ExtImm

        jal     Check_Parentheses_2

        addi    $t5, $zero, 10

        beq     $t9, $t5, Right_code

        addi    $t5, $zero, 0
        beq     $t9, $t5, Right_code    addi

$t5, $zero, 41                  #t5 ~ ')'       beq

$t9, $t5, Right_code

        j       False_code

#Check_Parentheses_1   Kiem tra dau ( Check_Parentheses_1:
        li      $t1, 0 #i = 0

        add     $a2, $s2, $t1
```

```
        lb      $t2, 0($a2)
        li      $t5, 40
        bne     $t2, $t5, False_code
        addi    $t1, $t1, 1
        add     $a2, $s2, $t1
        lb      $t2, 0($a2)
        bne     $zero, $t2, False_code
        jr      $ra

#Check_Parentheses_2   Kiem tra dau )
Check_Parentheses_2:
        li      $t1, 0 #i = 0
add     $a2, $s2, $t1         lb
$t2, 0($a2)

        li      $t5, 41
        bne     $t2, $t5, False_code
        addi    $t1, $t1, 1
        add     $a2, $s2, $t1
        lb      $t2, 0($a2)
        bne     $zero, $t2, False_code
        jr      $ra
#-------------------------------------------------
#Check Label Check_Label:
```

```
        li      $t1, 0          #i = 0
        add     $a2, $s2, $t1
        lb      $t2, 0($a2)
        beq     $t2, $zero, False_code
        li      $t5, 10                 #t5 = 'New line'
        beq     $t2, $t5, False_code
        slti    $t4, $t2, 48
        bne     $t4, $zero, False_code
        li      $t5, 58
        beq     $t2, $t5, False_code
        li      $t5, 59
        beq     $t2, $t5, False_code
        li      $t5, 60
        beq     $t2, $t5, False_code
        li      $t5, 61
        beq     $t2, $t5, False_code
        li      $t5, 62
        beq     $t2, $t5, False_code
        li      $t5, 63
        beq     $t2, $t5, False_code
        li      $t5, 64
        beq     $t2, $t5, False_code
        li      $t5, 91
```

```
        beq     $t2, $t5, False_code
        li      $t5, 92
        beq     $t2, $t5, False_code
        li      $t5, 93
        beq     $t2, $t5, False_code
        li      $t5, 94
        beq     $t2, $t5, False_code
        li      $t5, 96
        beq     $t2, $t5, False_code
        slti    $t4, $t2, 123
        beq     $t4, $zero, False_code
        addi $t1, $t1, 1 Loop_Label:

        add     $a2, $s2, $t1
        lb      $t2, 0($a2)
        beq     $t2, $zero, True_Label
        li      $t5, 10                 #t5 = 'New line'
        beq     $t2, $t5, True_Label
        slti    $t4, $t2, 48
        bne     $t4, $zero, False_code
        li      $t5, 58
        beq     $t2, $t5, False_code
        li      $t5, 59
```

```
        beq     $t2, $t5, False_code
        li      $t5, 60
        beq     $t2, $t5, False_code
        li      $t5, 61
        beq     $t2, $t5, False_code
        li      $t5, 62
        beq     $t2, $t5, False_code
        li      $t5, 63
        beq     $t2, $t5, False_code
        li      $t5, 64
        beq     $t2, $t5, False_code
        li      $t5, 91
        beq     $t2,            $t5,
                False_code
        li      $t5, 92
        beq     $t2, $t5, False_code
        li      $t5, 93
        beq     $t2, $t5, False_code
        li      $t5, 94
        beq     $t2, $t5, False_code
        li      $t5, 96
        beq     $t2, $t5, False_code
        slti    $t4, $t2, 123
```

```
        beq     $t4,            $zero,
        False_code
        addi    $t1, $t1, 1
        j       Loop_Label
#-------------------------------------------------
True_Label:
        jal     Print_Right_Label
        j       Right_code
#-------------------------------------------------  #-------------------------------------------------
-
#Output
False_opcode:
        #Print "Opcode"
        li $v0, 4
        la $a0, Message2
syscall
        nop
        #Print Opcode Input
li $v0, 4       add $a0,
$zero, $s2      syscall
        nop
        #Print "Khong hop
le!"    li $v0, 4       la $a0,
Message4        syscall
```

```
        nop
jal False_code  j
End_main
Right_opcode:
#Print "Opcode"     li
$v0, 4          la $a0,
Message2    syscall
        nop
        #Print Opcode Input
        li $v0, 4
        add $a0, $zero, $s2
syscall
        nop
        #Print ", hop le!"
li $v0, 4       la $a0,
Message3    syscall
        nop    jr
$ra
Right_Register:
        #Print "\n"     li
$v0, 4          la $a0,
Message7    syscall
        nop
```

```
        #Print "Thanh ghi"
li $v0, 4       la $a0,
Message8    syscall
        nop
        #Print Register Input
li $v0, 4
        add $a0, $zero, $s2
        syscall
        nop
        #Print ", hop le!"
li $v0, 4       la $a0,
Message3    syscall
        nop    jr
$ra
Print_Right_Number:
#Print "\n"    li $v0, 4
la $a0, Message7
syscall
        nop
        #Print "So "   li
$v0, 4      la $a0,
Message9    syscall
```

```
        nop
        #Print so trong hang
doi     li $v0, 4       add $a0,
$zero, $s2
        syscall
        nop
        #Print '', hop le!''
li $v0, 4       la $a0,
Message3    syscall
        nop     jr
$ra
Print_Right_Label:
        #Print ''\n''    li
$v0, 4          la $a0,
Message7    syscall
        nop
        #Print ''So ''   li
$v0, 4          la $a0,
Message10   syscall
        nop
        #Print label trong hang
doi     li $v0, 4       add $a0,
$zero, $s2    syscall
```

```
        nop
        #Print ", hop le!"
li $v0, 4       la $a0,
Message3    syscall
        nop    jr
$ra
Right_code:
        #Print "Right
code"       li $v0, 4
la $a0, Message5
syscall
        nop    j
End_main
False_code:
        #Print "False
code"       li $v0, 4
la $a0, Message6
syscall
        nop    j
End_main
End_main:
Run_Again: li $v0, 50
                la $a0, Message11
```

```
                syscall

                nop

                beq $a0, $zero, clear

                nop

                j exit

                nop

# clear: dua string ve trang thai ban dau de thuc hien lai qua trinh
clear:          add     $s3, $zero, $s1
Loop_Null:
                lb      $t3, 0($s3)

                li      $t5, 10
                beq     $t3, $t5, Loop_Null_them

                nop

                sb      $zero, 0($s3)

        addi  $s3, $s3, 1

                j       Loop_Null

Loop_Null_them:

                sb      $zero, 0($s3)

        j start

                nop

exit:li
     $v0,
     10
                syscall
```

Giải thuật:

-Để có thể check xem câu lệnh có đúng hay không thì ta cần lưu các opcode thỏa mãn, các thanh ghi thỏa mãn vào một chỗ nào đó trước rồi khi kiểm tra ta lại lấy ra để so sánh xem opcode nhập vào và thanh ghi nhập vào đã đúng chưa. Ở đây mình sẽ sử dụng mảng ký tự để lưu các opcode mẫu cũng như là các thanh ghi mẫu. và các từ cách nhau bởi dấu '/'. Và kết thúc xâu bởi dấu Space.

-Sau khi lưu các opcode, thanh ghi thỏa mãn như trên rồi thì ta sẽ dựng các hàm để tách và check riêng từng bộ phận của câu lệnh như: Opcode, thanh ghi, số, label, và kiểu dạng như 0($s2).

Sau khi tách được một phần thì ta phải đưa phần đấy đi kiểm tra xem có thỏa mãn hay không. Vậy rõ ràng ta cần phải có một vùng để lưu dữ liệu sau khi tách và đưa dữ liệu này đi kiểm tra. Ở đây mình sử dụng cấu trúc hàng đợi để kiểm soát vùng dữ liệu này. Địa chỉ đầu của hàng đợi được khai báo là 'chain_check' và được lưu vào s2 và vị trí phần tử cuối cùng của hàng đợi được lưu vào s0 (Lưu ý: ví dụ hàng đợi có lưu "beq" thì s0 = 2 vì phần tử đầu tiên được đánh số là 0).

-Và ứng với từng kiểu khuôn dạng (với từng giá trị s4) thì ta sẽ nhảy để hàm check tương ứng của khuôn dạng đó (khuôn dạng R, R1, I, ….)

-Sau khi check xong thì ta sẽ in ra kết quả rằng câu lệnh nhập vào có cấu trúc đúng hay sai.