

# BÁO CÁO KẾT QUẢ THỰC HÀNH

MÔN: KIẾN TRÚC MÁY TÍNH

KIỂM TRA GIỮA KỲ 2021/2

ĐINH THỊ THU HÀ – 20194543

## PART A:

### Assignment 6:

- Chương trình:

```
.data
nhapA: .asciiz "Nhap a:"
nhapB: .asciiz "Nhap b:"
nhapC: .asciiz "Nhap c:"
TG: .asciiz "a,b,c la 3 canh cua tam giac"
notTG: .asciiz "a,b,c khong la 3 canh cua tam giac"

.text
#### nhap a tu ban phim
li $v0, 51
la $a0, nhapA
syscall

#### luu a vao s0
addi $s0, $a0, 0

#### nhap b tu ban phim
li $v0, 51
la $a0, nhapB
syscall

#### luu b vao s1
addi $s1, $a0, 0

#### nhap c tu ban phim
li $v0, 51
la $a0, nhapC
syscall

#### luu c vao s2
addi $s2, $a0, 0

check1:
add $t0, $s0, $s1          # t0 = s0 + s1 = a + b
bgt $t0, $s2, check2       # neu (a+b) > c, check tiep
j notT_g                   # nguoc lai a,b,c khong phai 3 canh
nop

check2:
add $t0, $s0, $s2          # t0 = s0 + s2 = a + c
bgt $t0, $s1, check3       # neu (a+c) > b, check tiep
j notT_g                   # nguoc lai a,b,c khong phai 3 canh
nop

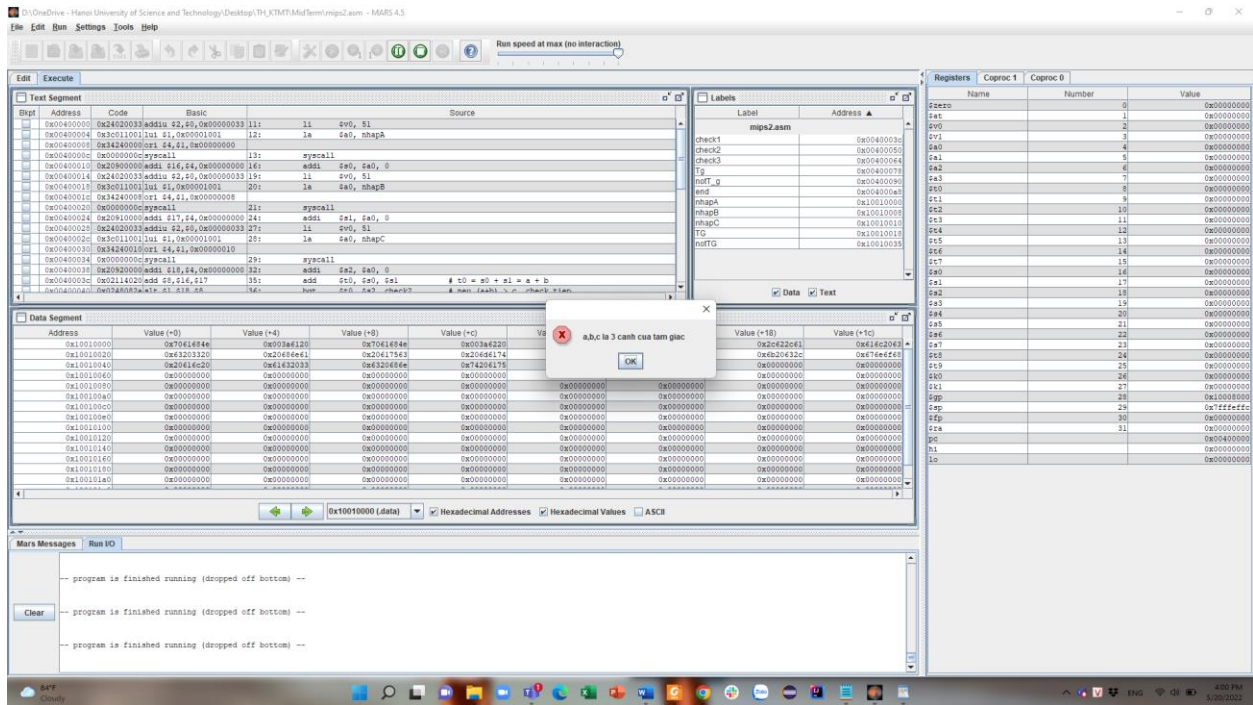
check3:
add $t0, $s1, $s2          # t0 = s1 + s2 = b + c
bgt $t0, $s0, Tg           # neu (b+c) > a ==> 3 canh tam giac
j notT_g                   # nguoc lai a,b,c khong phai 3 canh
nop

Tg:
#### in ra ket qua la tam giac
li $v0, 55
la $a0, TG
syscall
j end
nop

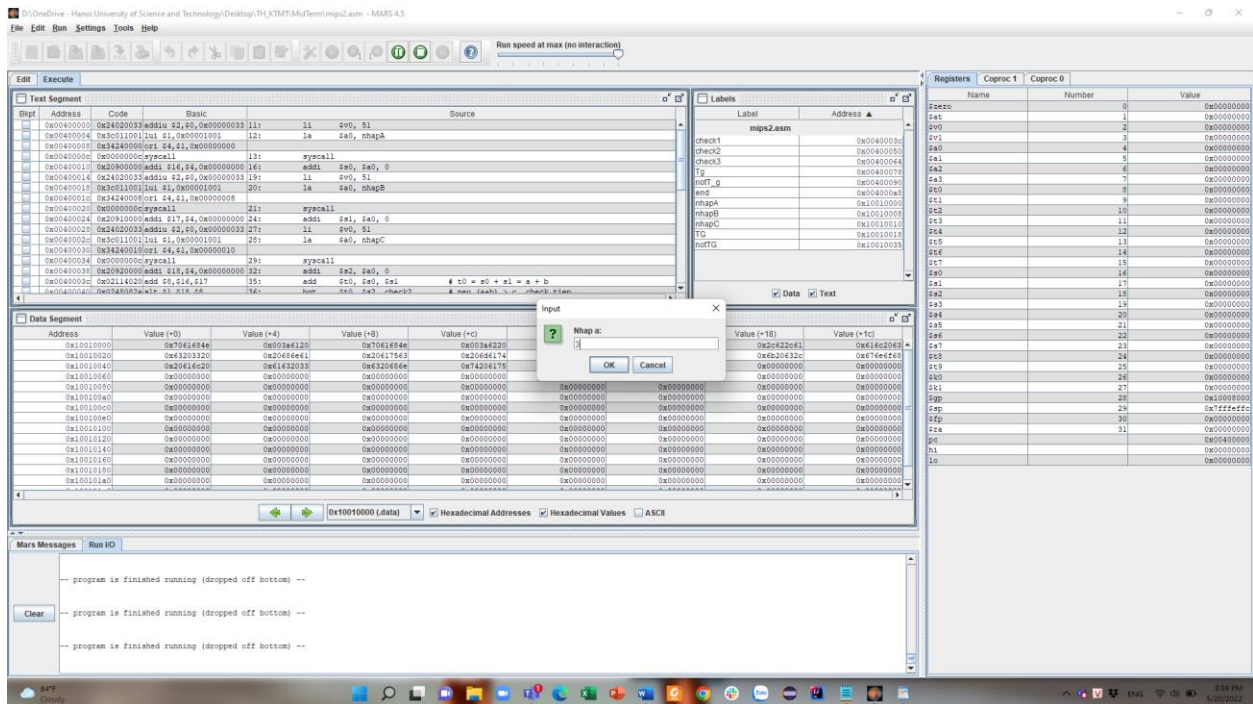
notT_g:
#### in ra ket qua khong la tam giac
li $v0, 55
la $a0, notTG
syscall
j end
nop

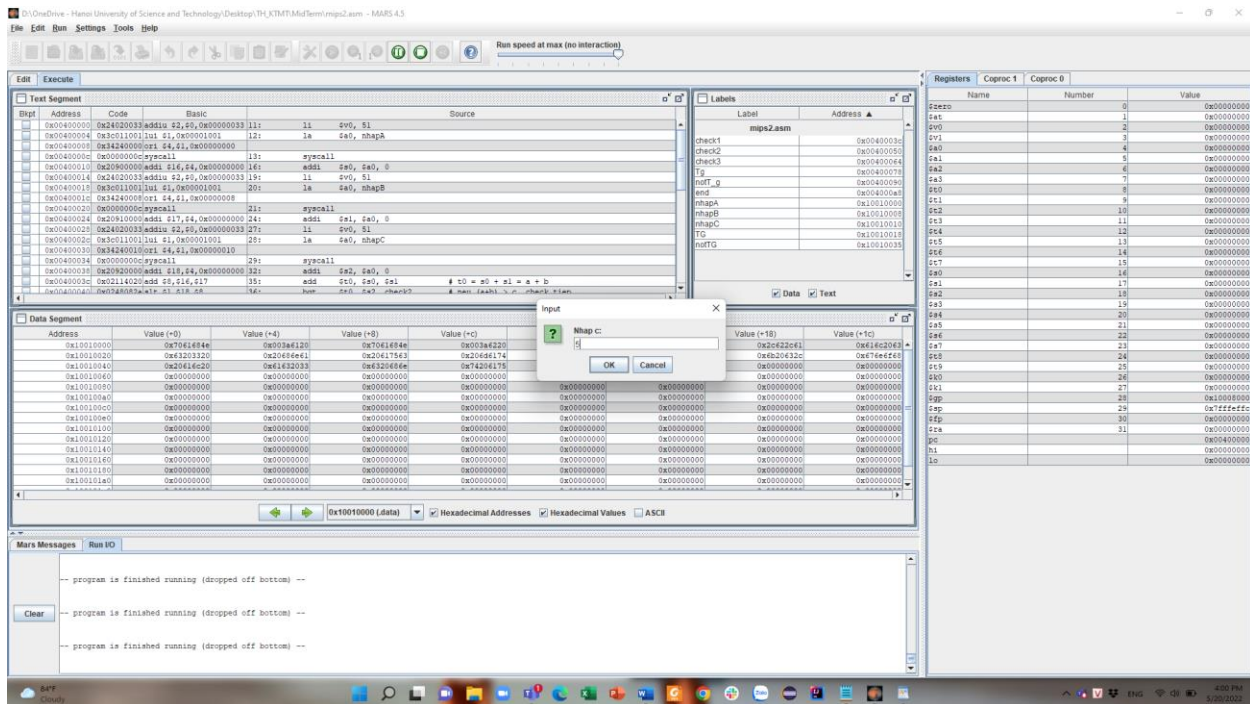
end:
```

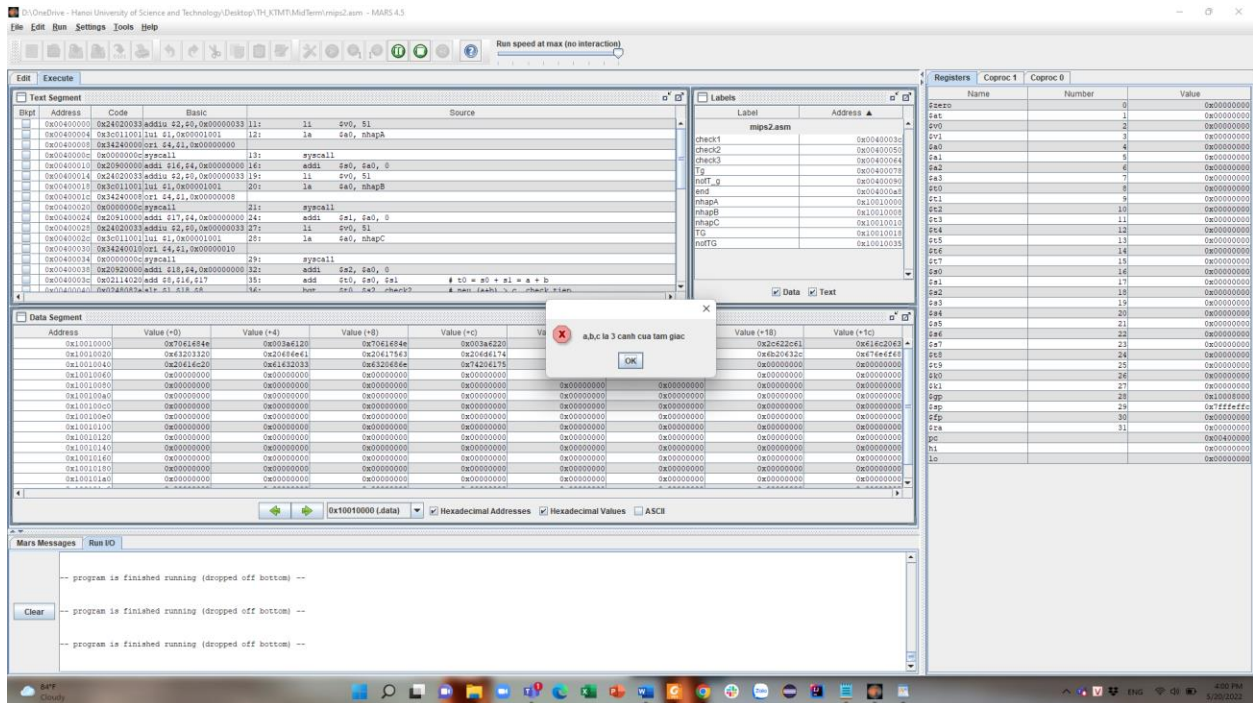
## - Kết quả:



## - Khi nhập 3 cạnh a,b,c lần lượt là 3,4,5 thì màn hình in ra là a,b,c là 3 cạnh của tam giác.







- **Phân tích cách thực hiện:**
  - + Điều kiện để 3 số là 3 cạnh của một tam giác là: Tổng của 2 cạnh bất kì luôn lớn hơn cạnh còn lại.(1)
  - + Nhập vào độ dài 3 cạnh a,b,c vào hộp thoại bằng cách gọi hàm `$v0=51`
  - + Sử dụng `check1`, `check2`, `check3` để kiểm tra (1).(kiểm tra nếu `check1` đúng mới đến `check2`, `check2` đúng mới đến `check3`)
  - + Nếu cả 3 check cùng đúng thì in ra kết quả "a,b,c la 3 canh cua tam giac"
  - + Nếu `check1` hoặc `check2` hoặc `check3` sai thì in ra kết quả là "a,b,c khong la 3 canh cua tam giac"
  - + Với hàm `check(check1, check2, check3)`: sau khi tính tổng 2 cạnh bất kì bằng lệnh `add`, sử dụng lệnh `bgt` để so sánh tổng lớn hơn cạnh còn lại. Nếu tổng lớn hơn thì nhảy đến check kế tiếp, ngược lại thì hiện kết quả là "a,b,c khong la 3 canh cua tam giac".

## PART B:

### Assignment 4:

- **Chương trình:**

```

.data
message_input_length:      .ascii "Nhập vào kích thước mảng: "
message_input_element:    .ascii "Nhập danh sách các phần tử (bấm enter giữa các phần tử): \n"
message_solved:           .ascii "Phần tử âm lớn nhất là: "
.text
input_length:  li    $v0, 4
               la    $a0, message_input_length
               syscall
               li    $v0, 5                #v0 chưa số nhập vào
               syscall

               add    $a1, $zero, $v0      #a1 chưa kích thước của mảng

input_element: li    $v0, 4
               la    $a0, message_input_element
               syscall

loop:          beq    $s1, $a1, end_loop
               li    $v0, 5                #nhập phần tử mảng
               syscall
               sge    $t4, $v0, $zero      # v0 > 0, t4 = 1, else t4 = 0
               bne    $t4, $zero, next_ele # t4 = 1, next_ele
               add    $a3, $zero, $v0      #khai tạo giá trị lớn nhất

next_ele:      sw     $v0, 0($sp)          #lưu giá trị vừa nhập vào A[i]
               add    $s1, $s1, 1
               sub    $sp, $sp, 4         #tro đến phần tử A[i+1]

               j      loop

end_loop:
main:          jal    maxNegative
               nop

               li    $v0, 4
               la    $a0, message_solved
               syscall

               li    $v0, 1
               add    $a0, $zero, $a3     #a3 chưa số nguyên âm lớn nhất
               syscall

end_main:      li    $v0, 10
               syscall

maxNegative:   add    $sp, $sp, 4          #đưa con trỏ về vị trí phần tử cuối cùng
               li    $s1, -1
               add    $a1, $a1, -1

maxNeg_loop:  beq    $s1, $a1, end_maxNeg
               add    $s1, $s1, 1
               add    $sp, $sp, 4          #tro đến phần tử A[i-1]
               lw     $t1, 0($sp)         #lưu giá trị A[i] vào t1
               sge    $t2, $t1, $zero      #nếu t1 < 0
               bne    $t2, $zero, maxNeg_loop
               sgt    $t3, $t1, $a3        #nếu t1 > a3
               beq    $t3, $zero, maxNeg_loop
               add    $a3, $zero, $t1
               j      maxNeg_loop

end_maxNeg:    jr     $ra

```

- **Kết quả:**

+ Trường hợp trong mảng nhập vào có số âm và số dương, kết quả sẽ trả về là số âm lớn nhất.

```

Mars Messages Run IO
Nhập vào kích thước mảng: 5
Nhập danh sách các phần tử (bấm enter giữa các phần tử):
-3
-5
1
7
-20
Phần tử âm lớn nhất là: -3
-- program is finished running --

```

+ Trường hợp trong mảng nhập vào chỉ có số dương hoặc cả số dương và số 0, kết quả sẽ trả về là 0.



```

Nhap vao kich thuc mang: 4
Nhap danh sach cac phan tu (bam enter giua cac phan tu):
1
2
3
4
Phan tu am lon nhat la: 0
-- program is finished running --

```

+ Trường hợp trong mảng nhập vào có số 0, vì số 0 không phải là nguyên dương cũng không phải số nguyên âm nên kết quả trả về số âm lớn nhất khác 0.

```

Nhap vao kich thuc mang: 3
Nhap danh sach cac phan tu (bam enter giua cac phan tu):
0
1
-3
Phan tu am lon nhat la: -3
-- program is finished running --

```

- *Phân tích cách thực hiện:*

## PART C:

### Assignment 1:

- *Chương trình:*

```

.data

String:      .space 40
result:      .space 40

space:       .asciiiz " "
nhapXau:     .asciiiz "Nhap vao xau (do dai max 40) : "
res:         .asciiiz "Tu ngan nhat trong xau la: "

.text

main:
    jal      input
    jal      process
    j        printResult

input:
    li       $v0, 54
    la       $a0, nhapXau
    la       $a1, String          # a1 luu String nhap vao
    li       $a2, 40              # do dai max cua String
    syscall

### in ra xau vua nhap
    li       $v0, 4
    la       $a0, String
    syscall

    jr       $ra

process:
    addi     $fp, $sp, 0          # luu sp ban dau
    la       $s0, String          # dia chi cua xau String
    addi     $t0, $s0, 0          # t0 tro den dia chi String[0]
    addi     $t1, $zero, 0        # do dai dang xet
    addi     $s1, $zero, 0        # do dai max

```

```

loop:
    lb      $t2, 0($t0)          # t1 = String[i]
    li      $t7, 0              # t7 = 0, danh dau thoat lap
    beq     $t2, '\0', check     # s[i] = "\0", check lan cuoi va thoat lap
    li      $t7, 1              # t7 = 1, tiep tục vòng lặp
    beq     $t2, ' ', check      # gap dau cach " ", duyet xong 1 tu, check
    beq     $t2, '\n', continue  # gap '\n' thi skip

checkWord:
    addi    $t1, $t1, 1          # do dai dang xet + 1
    sb      $t2, 0($sp)          # luu ky tu String[i] vao stack
    addi    $sp, $sp, -1         # sp = sp - 1, them 1 byte vao ngan nho stack
    j       continue

check:
    beqz    $s1, update          # s1 chua co gia tri, cap nhat vao result xau vua xet
    blt     $t1, $s1, update     # t1 < s1, update
    j       reset               # t1 >= s1, reset va chay tiep vòng lặp

update:
    addi    $t4, $sp, 0          # luu dia chi stack hien tai
    addi    $sp, $fp, 0          # lay dia chi duoi cung cua stack
    la      $t3, result          # t3 chua dia chi xau result

getString:
    lb      $t5, 0($sp)          # t5 = gia tri o stack
    sb      $t5, 0($t3)          # result[i] = t5

    addi    $sp, $sp, -1         # tro den gia tri stack ben tren
    addi    $t3, $t3, 1          # tro den result[i+1]
    bne     $t4, $sp, getString  # xet cac ky tu cua tu vua luu

reset:
    li      $t8, '\0'           # them '\0' vao cuoi result
    sb      $t8, 0($t3)          # update gia tri cua do dai max
    addi    $s1, $t1, 0          # tro ve dau stack
    addi    $sp, $fp, 0
    addi    $t1, $zero, 0

continue:
    addi    $t0, $t0, 1          # tro den dia chi String[i+1]
    beqz    $t7, endloop         # t7 = 0, thoat lap
    j       loop

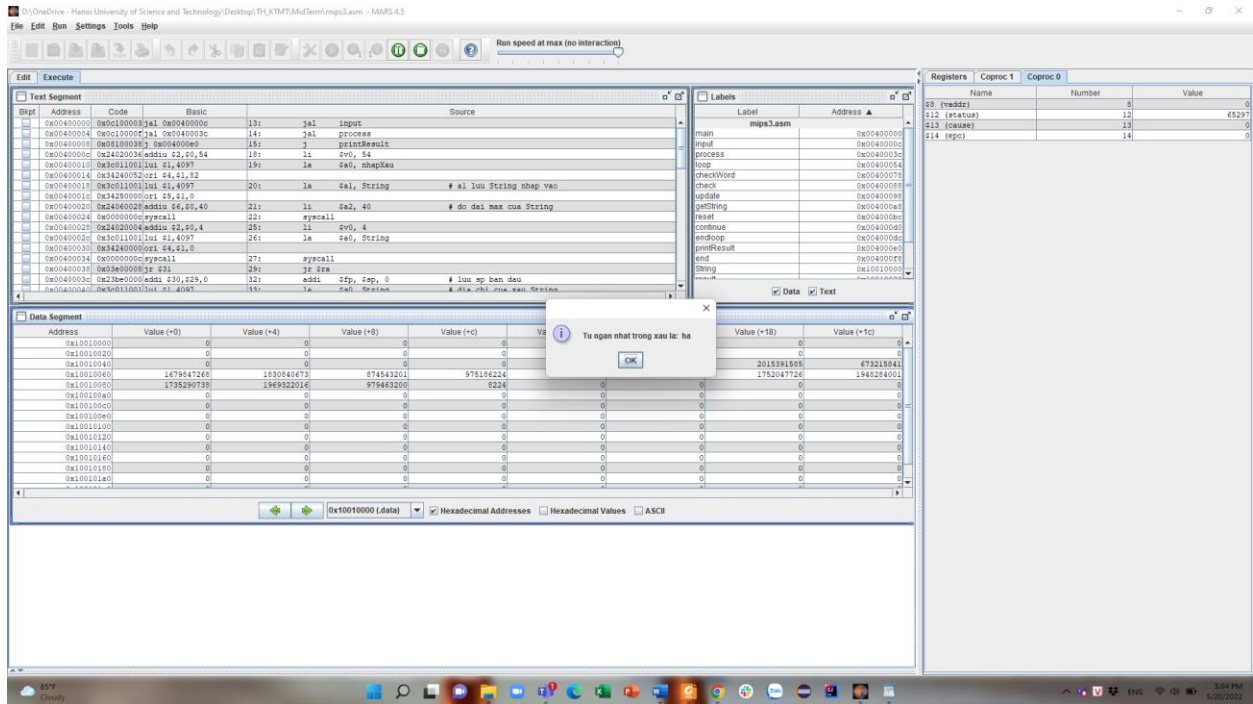
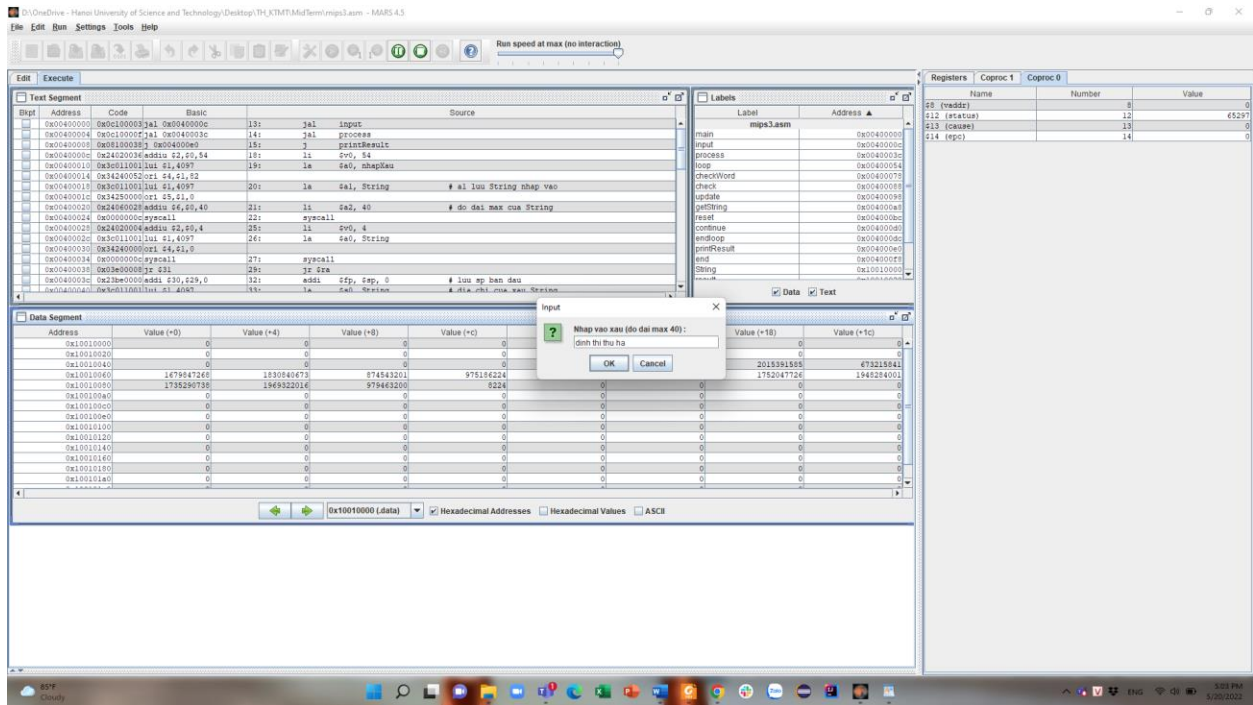
endloop:
    jr      $ra

#####3 in ket qua tu ngan nhât tìm được
printResult:
    li      $v0, 59
    la      $a0, res
    la      $a1, result
    syscall

end:

```

- **Kết quả:**



- Phân tích cách thực hiện:

+ Nhập vào xâu kí tự (dưới 40 ký tự)

+ Kiểm tra trong xâu ký tự từ nào ngăn nhất và in ra màn hình. (Mỗi dấu cách là một từ).