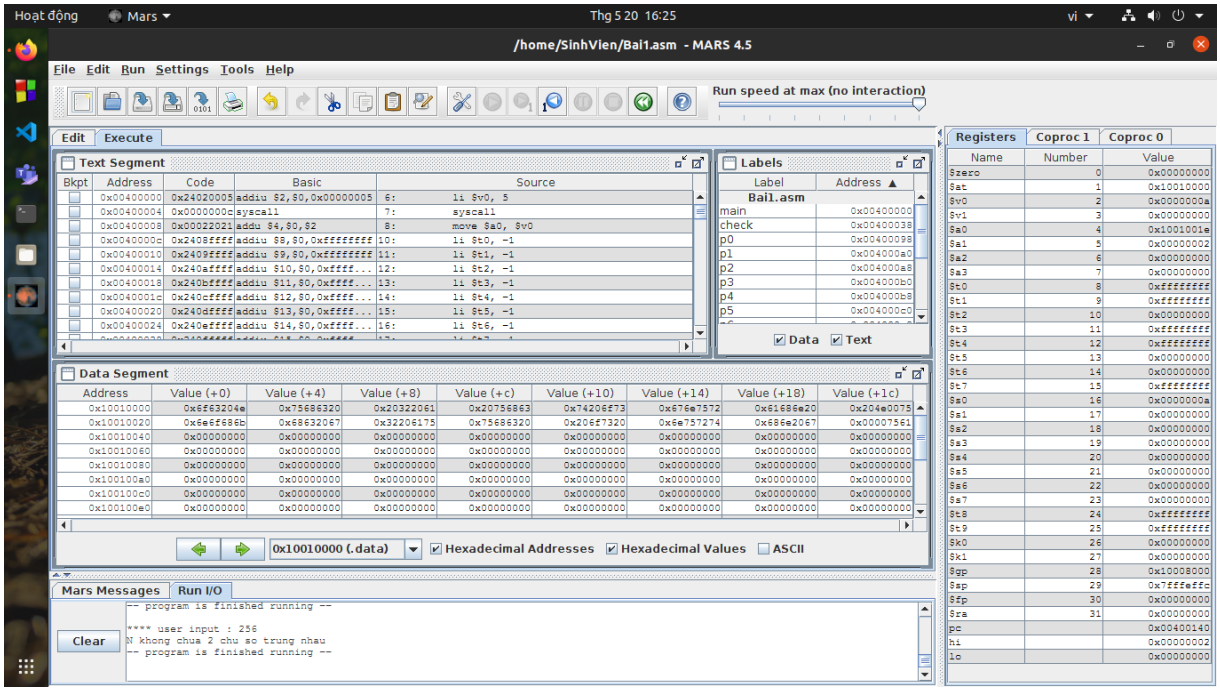
Bài 1: (A3)

1. Ý tưởng

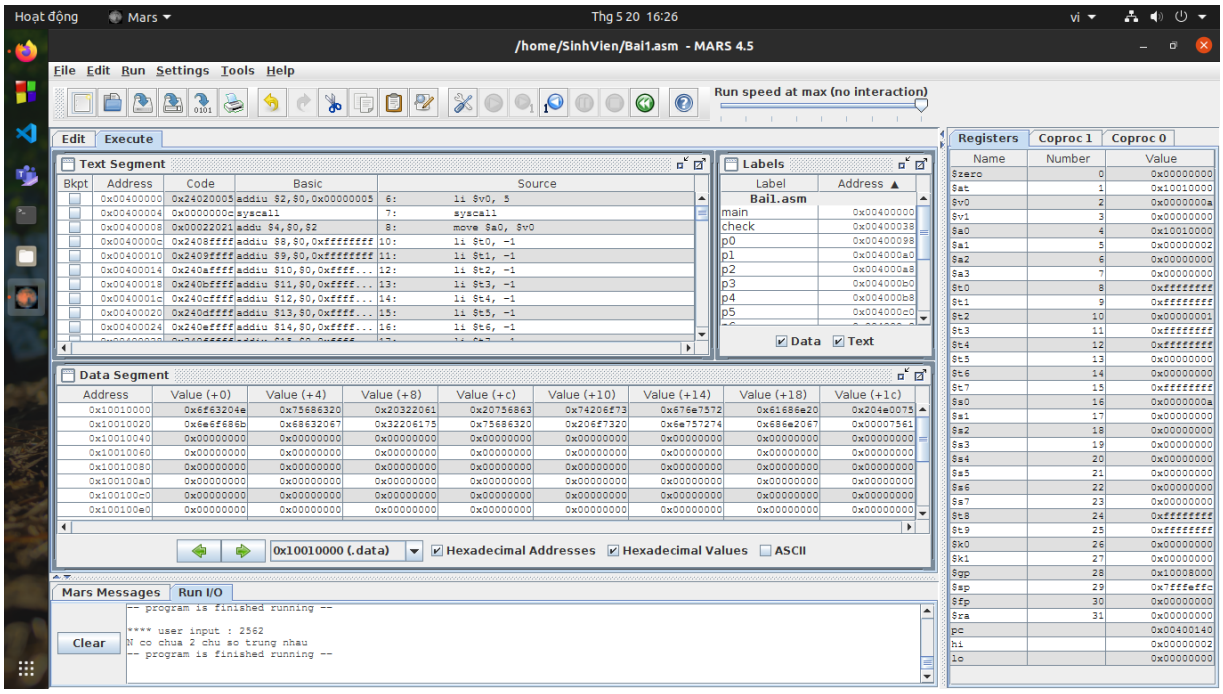
- Đầu tiên, em khởi tạo giá trị các thanh ghi từ $t0 đến $t9 là -1.
- Em dùng phép chia có dư cho 10 để lấy lần lượt từng chữ số cuối của số nhập vào. Với mỗi số thì em cộng thêm 1 vào thanh ghi tương ứng (ví dụ 256 chia 10 dư 6, thương là 25 => Em cộng 1 vào $t6, rồi tiếp tục vòng lặp, lấy 25 chia 10...)
- Từ đó, em lấy được tất cả các chữ số của số nhập vào, và nếu nó có số trùng nhau (ví dụ 2 xuất hiện 3 lần) thì thanh ghi $t2 = -1+3=2.
- Sau đó, em check xem nếu có 1 thanh ghi > 0 (nghĩa là được cộng 2 lần) thì có chữ số trùng nhau. Không thì không có.

2. Ảnh chụp chạy thử

- Khi nhập số 256 => N không chứa 2 chữ số trùng nhau (đúng)



- Khi nhập 2562 => N có chứa 2 chữ số trùng nhau (đúng)



3. Source code

.data

msg1: .asciiz "N co chua 2 chu so trung nhau"

msg2: .asciiz "N khong chua 2 chu so trung nhau"

.text

main:

li $v0, 5

syscall

move $a0, $v0

li $t0, -1

li $t1, -1

li $t2, -1

li $t3, -1

```
li $t4, -1
li $t5, -1
li $t6, -1
li $t7, -1
li $t8, -1
li $t9, -1

li $s0, 10

check:
beqz $a0, complete
div $a0, $s0
mfhi $a1    # remainder
mflo $a0    # quotient

beq $a1, 0, p0
beq $a1, 1, p1
beq $a1, 2, p2
beq $a1, 3, p3
beq $a1, 4, p4
beq $a1, 5, p5
beq $a1, 6, p6
beq $a1, 7, p7
beq $a1, 8, p8
```

```
beq $a1, 9, p9


p0:
add $t0, $t0, 1
j check
p1:
add $t1, $t1, 1
j check
p2:
add $t2, $t2, 1
j check
p3:
add $t3, $t3, 1
j check
p4:
add $t4, $t4, 1
j check
p5:
add $t5, $t5, 1
j check
p6:
add $t6, $t6, 1
j check
p7:
```

```
add $t7, $t7, 1
j check
p8:
add $t8, $t8, 1
j check
p9:
add $t9, $t9, 1
j check

complete:
bgtz $t0, trung
bgtz $t1, trung
bgtz $t2, trung
bgtz $t3, trung
bgtz $t4, trung
bgtz $t5, trung
bgtz $t6, trung
bgtz $t7, trung
bgtz $t8, trung
bgtz $t9, trung

la $a0, msg2
li $v0, 4
syscall
```

```
b exit

trung:
la $a0, msg1
li $v0, 4
syscall
b exit
exit:
li $v0, 10
Syscall
```

Bài 2: (B3)

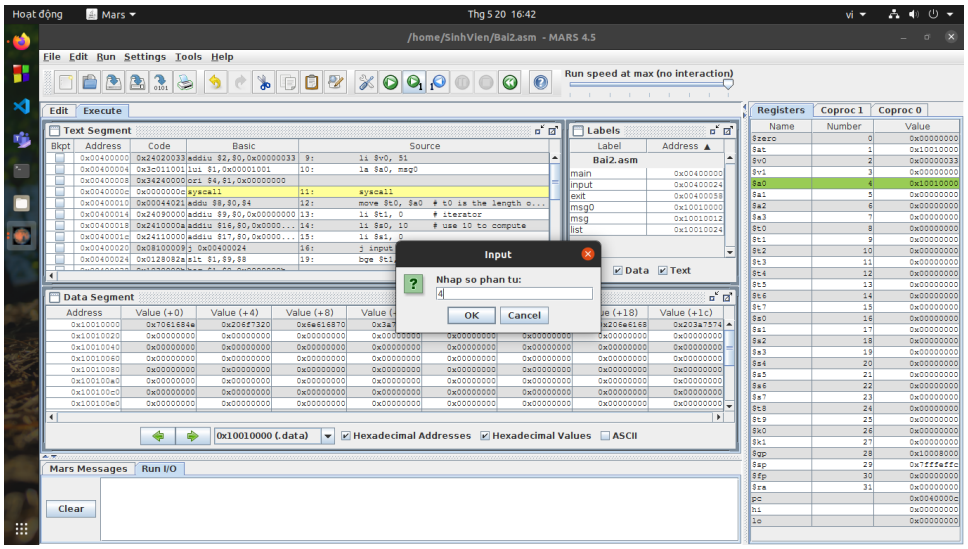
1. Ý tưởng

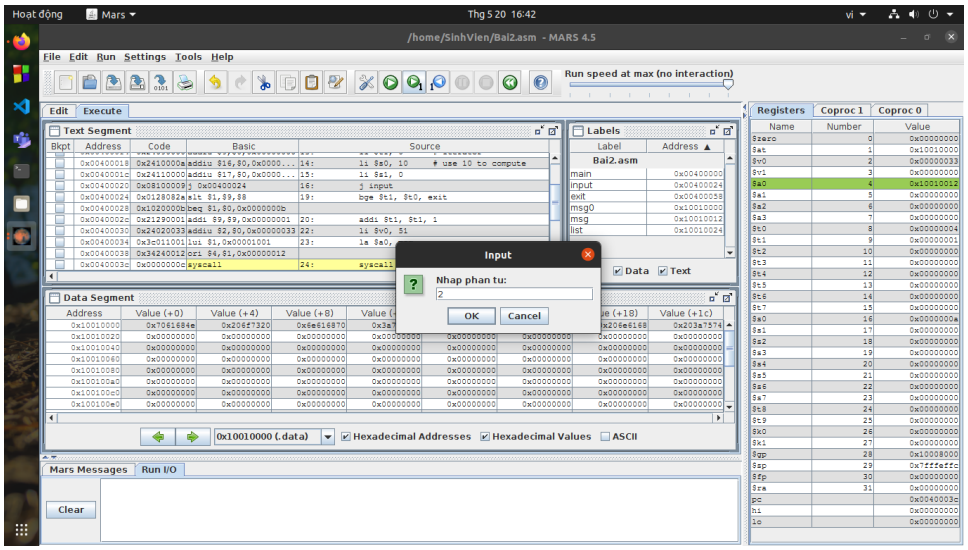
- Em tạo biến $s1 để tính tổng cần tìm, và mỗi khi nhập 1 phần tử mới ở mảng, em sẽ kiểm tra điều kiện của nó. Nếu thỏa mãn điều kiện thì sẽ cộng $s1 với phần tử mới đó, nếu không thì bỏ qua. Tiếp tục chạy vòng lặp đến khi nhập hết số lượng phần tử

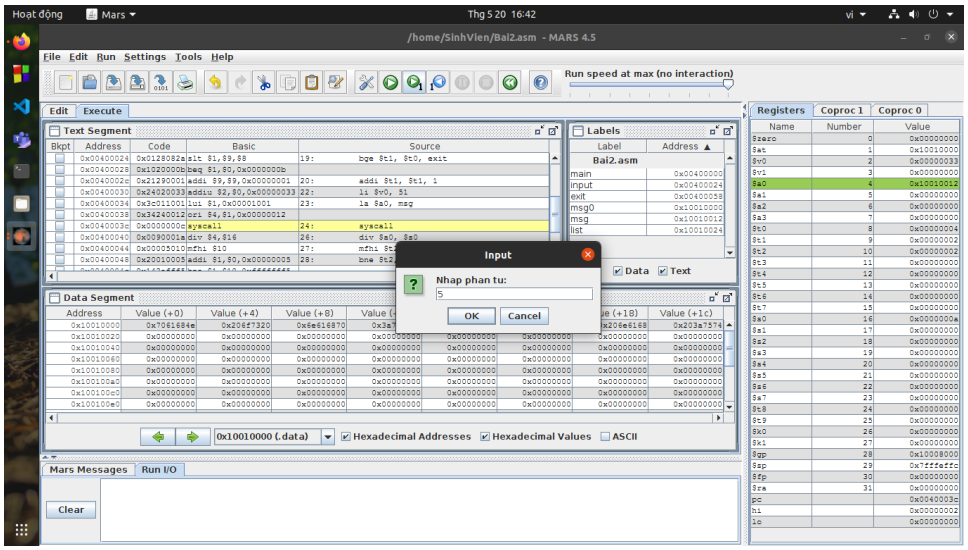
- “Điều kiện" ở ý trên là chia 10 dư 5. Dễ thấy tất cả các số lẻ chia hết cho 5 thì đều chia 10 dư 5 và ngược lại

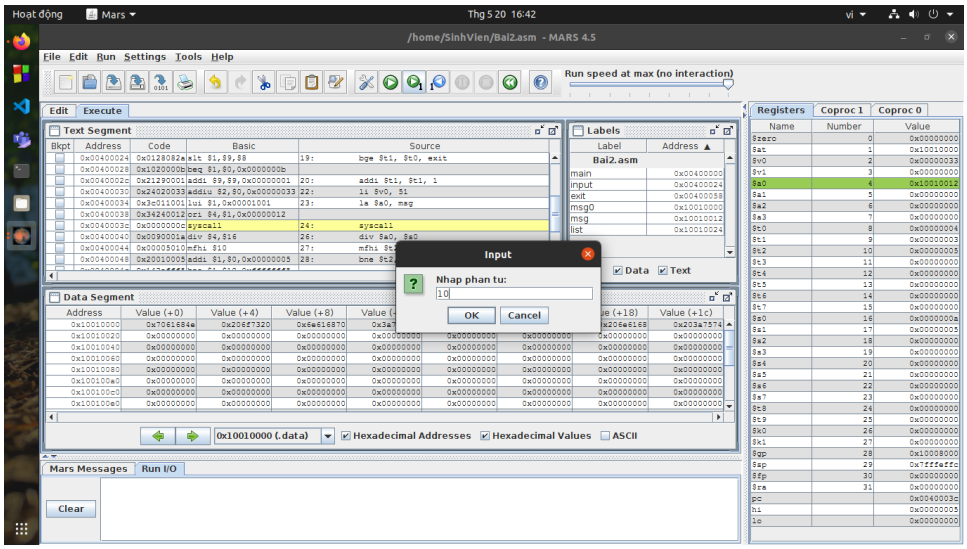
2. Ảnh chụp chạy thử

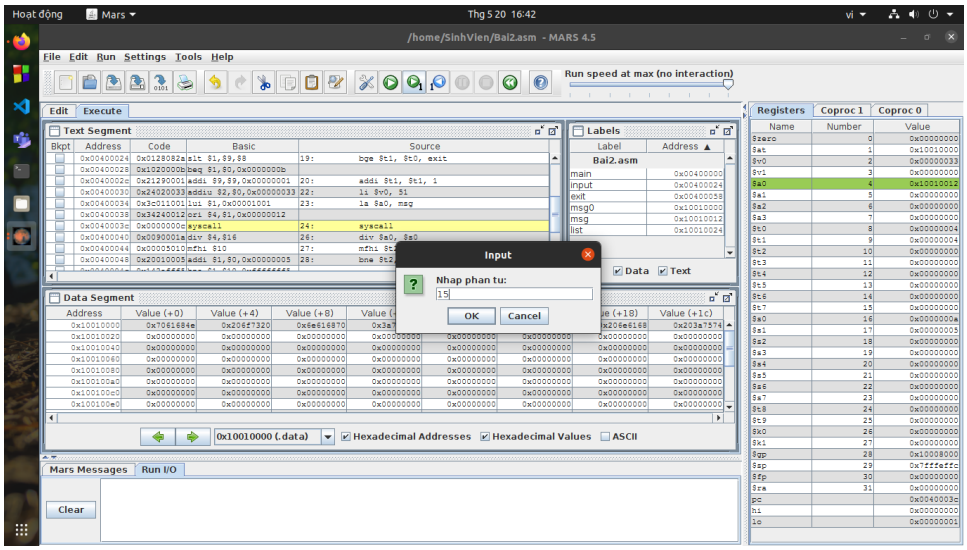
- Em nhập 4 số 2, 5, 10, 15 => Trả về 20 (đúng do 5+15=20)

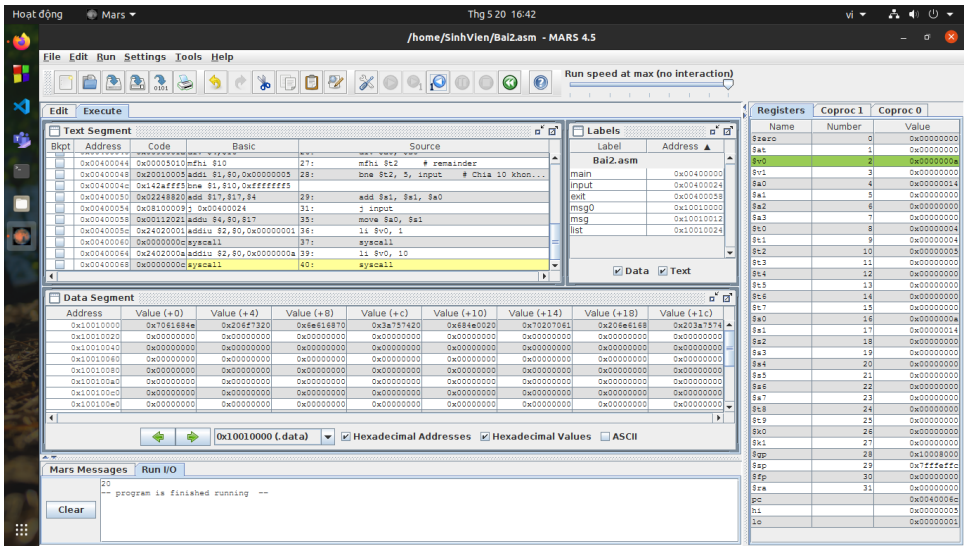

Hoạt động
Mars
Thg 5 20 16:42
vi
/home/SinhVien/Bai2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Registers Coproc 1 Coproc 0
Input
Nhap so phan tu:
4
OK Cancel
Data Segment
Mars Messages Run I/O
Clear


Hoạt động
Mars
Thg 5 20 16:42
vi
/home/SinhVien/Bai2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Registers Coproc 1 Coproc 0
Input
Nhap phan tu:
2
OK Cancel
Data Segment
Mars Messages Run I/O
Clear


Hoạt động
Mars
Thg 5 20 16:42
vi
/home/SinhVien/Bai2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Registers Coproc 1 Coproc 0
Input
Nhap phan tu:
5
OK Cancel
Data Segment
Mars Messages Run I/O
Clear


Hoạt động
Mars
Thg 5 20 16:42
vi
/home/SinhVien/Bai2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Registers Coproc 1 Coproc 0
Input
Nhap phan tu:
10
OK
Cancel
Data Segment
Mars Messages Run I/O
Clear


Hoạt động
Mars
Thg 5 20 16:42
vi
/home/SinhVien/Bai2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Registers Coproc 1 Coproc 0
Input
Nhap phan tu:
15
OK
Cancel
Data Segment
Mars Messages Run I/O
Clear


Hoạt động
Mars
Thg 5 20 16:42
vi
/home/SinhVien/Bai2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Registers Coproc 1 Coproc 0
Data Segment
Mars Messages Run I/O
20
-- program is finished running --
Clear

3. Source code

```
.data
msg0: .asciiz "Nhap so phan tu: "
msg: .asciiz "Nhap phan tu: "
.align 2
list: .space 100
.text
main:
# input len of array
li $v0, 51
la $a0, msg0
syscall
move $t0, $a0  # t0 is the length of array
li $t1, 0      # iterator
li $s0, 10     # use 10 to compute
li $s1, 0
j input

input:
bge $t1, $t0, exit
addi $t1, $t1, 1
# Nhap phan tu
li $v0, 51
la $a0, msg
```

syscall

div $a0, $s0

mfhi $t2 # remainder

bne $t2, 5, input # Chia 10 khong du 5 => khong thoa man

add $s1, $s1, $a0

j input

exit:

 # Print the sum

move $a0, $s1

li $v0, 1

syscall

li $v0, 10

Syscall

Bài 3: (C6)

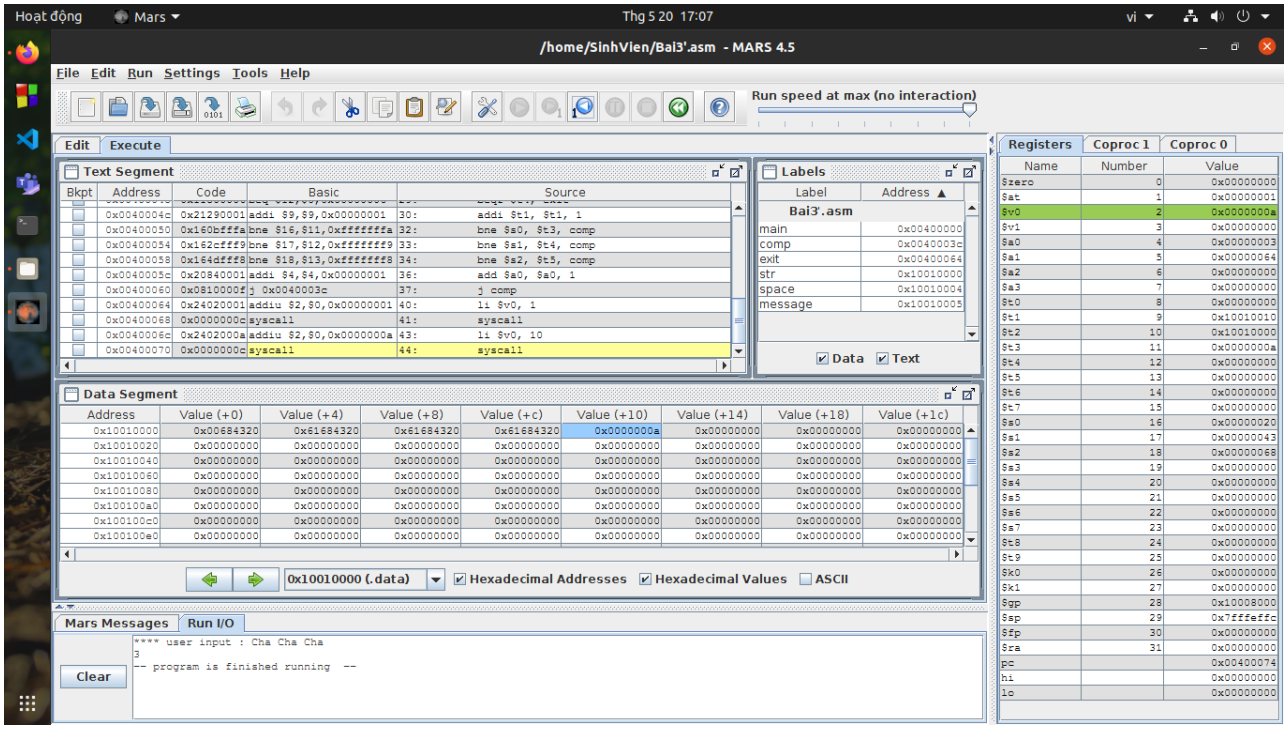
1. Ý tưởng:

- Do từ bắt đầu bằng “Ch" chứ không phải “Ch" ở giữa 1 từ, em đã quét lần lượt 3 ký tự liên tiếp từ đầu đến cuối của xâu ký tự. Nếu 3 ký tự đó giống với “ Ch" thì là có thêm 1 từ trong xâu bắt đầu bằng “Ch"

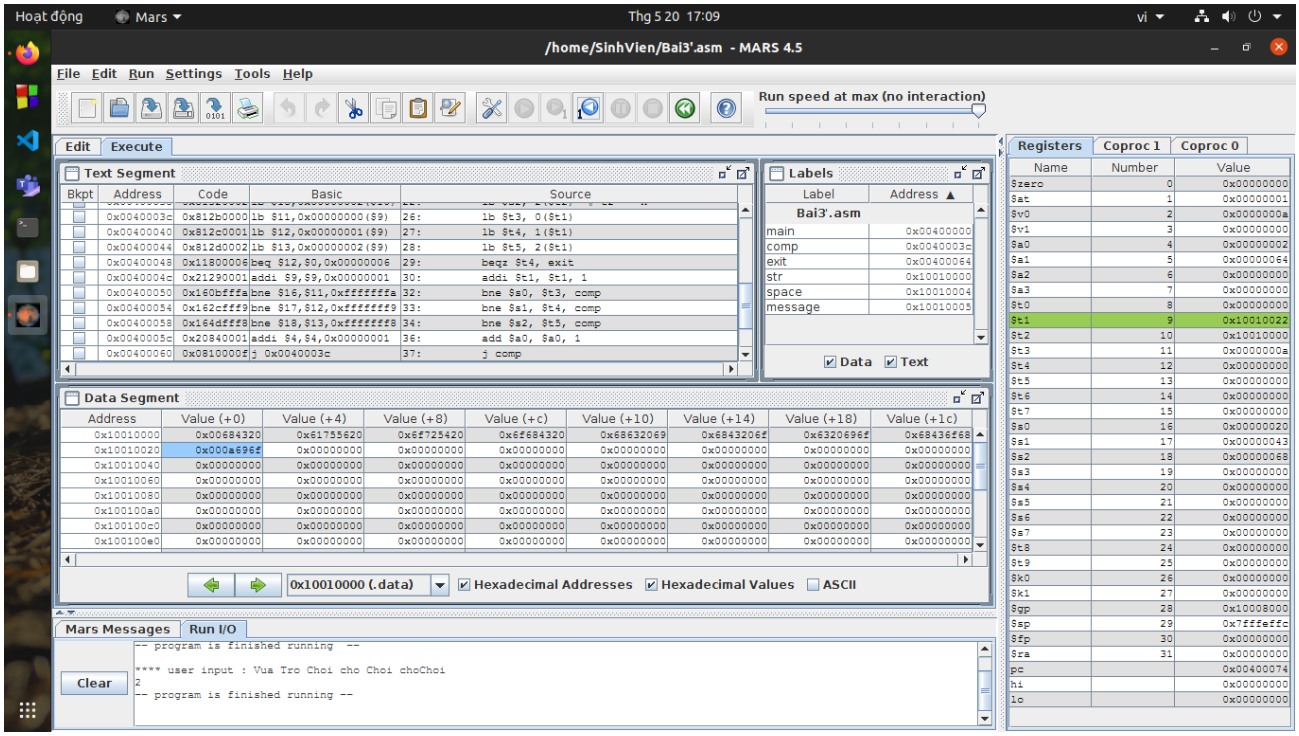
- Để so sánh 2 chuỗi có độ dài 3, em so sánh đôi một với nhau, nếu cả 3 ký tự đều giống nhau thì 2 chuỗi 3 ký tự đó giống nhau

2. Ảnh chụp chạy thử

- Em nhập “Cha Cha Cha" => Trả về 3 (đúng)



- Em nhập “Vua Tro Choi cho Choi choChoi” => Trả về 2 (đúng)



3. Source code

.data

str: .asciiz " Ch"

space: .ascii " "

message: .space 100

.text

main:

 # Read the string S from the user

la $a0, message

li $a1, 100

li $v0, 8

syscall

```
la $t1, message
la $t2, str
subi $t1, $t1, 1

li $a0, 0     # sum

lb $s0, 0($t2)  # t0 = " "
lb $s1, 1($t2)  # t1 = "C"
lb $s2, 2($t2)  # t2 = "h"

comp:
# load 3 character of the message continuously
lb $t3, 0($t1)
lb $t4, 1($t1)
lb $t5, 2($t1)
beqz $t4, exit
addi $t1, $t1, 1

bne $s0, $t3, comp
bne $s1, $t4, comp
bne $s2, $t5, comp

add $a0, $a0, 1
j comp
```

```
exit:
li $v0, 1
syscall

li $v0, 10
syscall
```