

Tên: Trần Hoài Phú

MSSV: 22521106

Lớp: IT006.O19 – **Thầy:** Ngô Hiếu Trường

BÀI TẬP ASSEMBLY

Bài 1: In chuỗi đảo ngược do người dùng nhập (tối thiểu 20 kí tự gồm hoa thường số) và viết hoa vị trí chẵn chuỗi bắt đầu từ số 0, 1, 2, 3, 4

#Code Assembly trên Mars

.data

TB_nhập : .asciiz "Nhập vào một chuỗi: "

TB_xuất: .asciiz "Chuỗi đã đảo ngược là: "

TB_độ_dài: .asciiz "Chuỗi ít nhất phải có 20 ký tự."

nhập: .space 100

xuất: .space 100

ĐK_dòng: .asciiz "\n"

.text

main:

In thông báo yêu cầu nhập chuỗi

li \$v0, 4

la \$a0, TB_nhập

syscall

Nhập chuỗi

li \$v0, 8

la \$a0, nhập

li \$a1, 100

syscall

Tính độ dài của chuỗi

```
li $t0, 0      # Số ký tự hiện tại
la $t1, nhap   # Địa chỉ bắt đầu của chuỗi
```

count_length:

```
lb $t2, 0($t1) # Đọc ký tự từ bộ nhớ
beqz $t2, end_count # Nếu ký tự là NULL (\0) thì kết thúc
addi $t1, $t1, 1 # Tăng địa chỉ vùng nhớ lên 1 byte
addi $t0, $t0, 1 # Tăng biến đếm lên 1
j count_length  # Quay lại vòng lặp
```

end_count:

```
# Kiểm tra nếu độ dài chuỗi ít nhất 20 ký tự
bge $t0, 20, reverse_string # Nếu có ít nhất 20 ký tự, chuyển đến phần đảo ngược chuỗi
li $v0, 4
la $a0, TB_too_short
syscall
j end_program
```

reverse_string:

```
# Tìm chiều dài chuỗi
li $t1, 0
la $t0, nhap
lb $t2, DK_dung
add $t3, $t0, $t1
lb $t4, 0($t3)
```

```
# In thông báo xuất chuỗi đảo ngược
li $v0, 4
la $a0, TB_xuat
syscall
```

loop_1:

```
beq $t4, $t2, exit_1
addi $t1, $t1, 1
add $t3, $t1, $t0
lb $t4, 0($t3)
j loop_1
```

exit_1:

```
move $s1, $t1
la $t5, xuat
li $t6, 0
add $t7, $t5, $t6
lb $s2, 0($t7)
```

loop_2:

```
beqz $t1, exit_2
subi $t1, $t1, 1
add $t3, $t1, $t0
lb $t4, 0($t3)
move $s2, $t4
sb $s2, 0($t7)
addi $t6, $t6, 1
add $t7, $t5, $t6
lb $s2, 0($t7)
j loop_2
```

exit_2:

```
la $t0, xuat
add $t2, $t0, $t1
lb $t3, 0($t2)
li $t8, 2
```

loop_3:

```
    beq $s1, $t1, exit_3
    div $t1, $t8
    mfhi $t4
    bnez $t4, ky_tu_ke_tiep
    blt $t3, 97, ky_tu_ke_tiep
    bgt $t3, 122, ky_tu_ke_tiep
    subi $t3, $t3, 32
    sb $t3, 0($t2)
    j ky_tu_ke_tiep
```

exit_3:

```
    move $a0, $t0
    li $v0, 4
    syscall
```

```
    li $v0, 10
    syscall
```

ky_tu_ke_tiep:

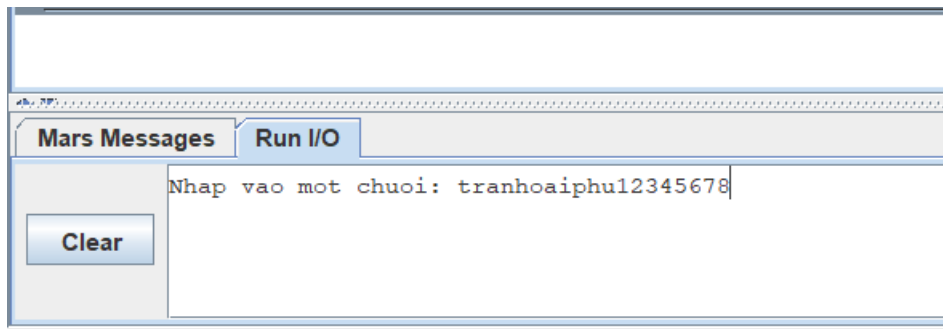
```
    addi $t1, $t1, 1
    add $t2, $t1, $t0
    lb $t3, 0($t2)
    j loop_3
```

end_program:

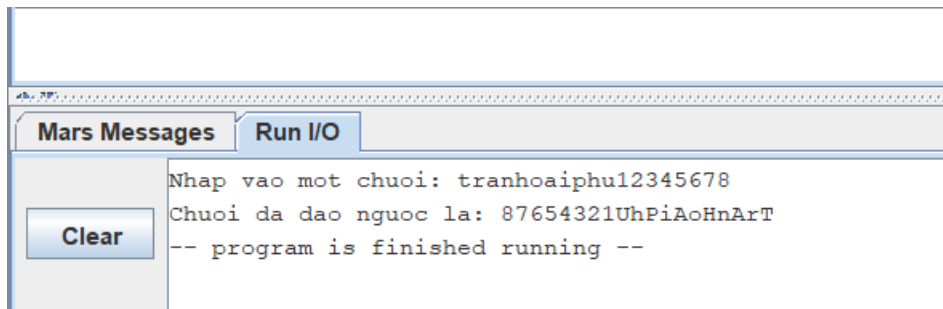
```
    # Kết thúc chương trình
    li $v0, 10
    syscall
```

#Chạy chương trình

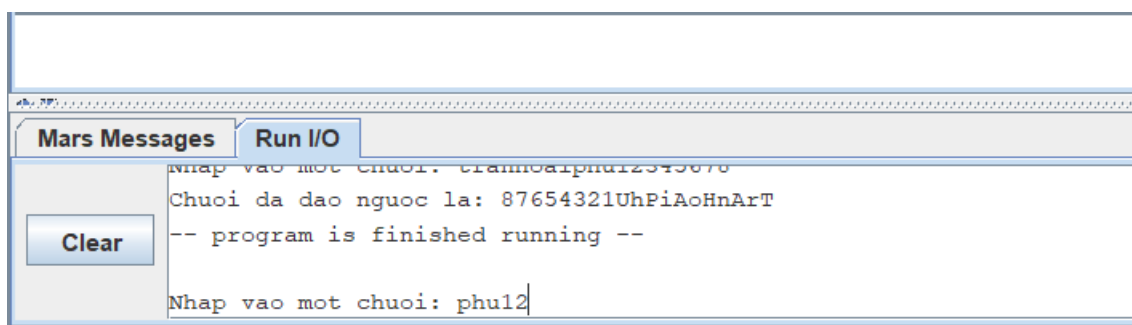
-Khi nhập vào chuỗi tranhoaiphu12345678



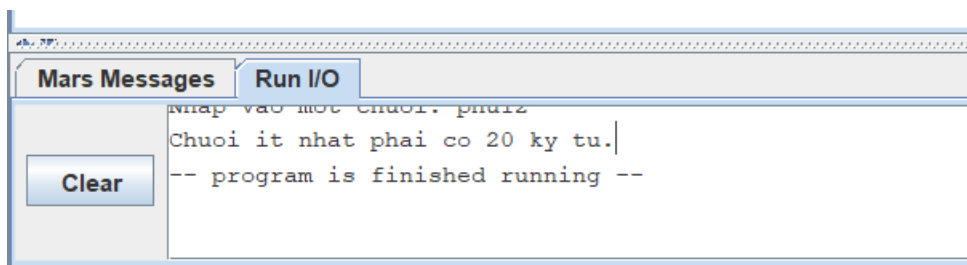
-Đáp án ra là một chuỗi đảo ngược và in hoa vị trí chẵn như hình “87654321UhPiAoHnArT”



-Khi nhập chuỗi là: “phu12”



-Đáp án xuất ra là : “Chuoi it nhat phai co 20 ky tu.” Đúng theo yêu cầu của đề bài là chuỗi phải có tối thiểu 20 ký tự



Bài 2: Đếm trong chuỗi vừa nhập bao nhiêu chữ hoa, chữ thường và số

#Code Assembly trên Mars

.data

string: .space 100

```
prompt: .asciiiz "\nNhap chuoi: "
result_lowercase: .asciiiz "\nSo chu cai thuong: "
result_uppercase: .asciiiz "\nSo chu cai hoa: "
result_digit: .asciiiz "\nSo chu so: "
```

```
.text
```

```
main:
```

```
    # In lời nhắc nhập chuỗi
```

```
    li $v0, 4
```

```
    la $a0, prompt
```

```
    syscall
```

```
    # Nhận chuỗi từ người dùng
```

```
    li $v0, 8
```

```
    la $a0, string
```

```
    li $a1, 100
```

```
    syscall
```

```
    # Khởi tạo biến đếm
```

```
    li $t0, 0    # Đếm chữ thường
```

```
    li $t1, 0    # Đếm chữ hoa
```

```
    li $t2, -1    # Đếm số
```

```
    la $t3, string # Địa chỉ của chuỗi
```

```
count_loop:
```

```
    lb $t4, 0($t3) # Lấy kí tự tiếp theo từ chuỗi
```

```
    beqz $t4, end_count # Nếu kí tự là NULL, dừng đếm
```

```
    # Kiểm tra và đếm số
```

```
    li $t5, 48    # Mã ASCII '0'
```

```
    li $t6, 57    # Mã ASCII '9'
```

ble \$t4, \$t6, add_so

b check_hoa

check_hoa:

Kiểm tra và đếm chữ hoa

li \$t5, 65 # Mã ASCII 'A'

li \$t6, 90 # Mã ASCII 'Z'

ble \$t4, \$t6, add_hoa

b check_thuong

check_thuong:

Kiểm tra và đếm chữ thường

li \$t5, 97 # Mã ASCII 'a'

li \$t6, 122 # Mã ASCII 'z'

ble \$t4, \$t6, add_thuong

b next_check

add_so:

addi \$t2, \$t2, 1

j next_check

add_hoa:

addi \$t1, \$t1, 1

j next_check

add_thuong:

addi \$t0, \$t0, 1

addi \$t3, \$t3, 1

j count_loop

next_check:

```
addi $t3, $t3, 1    # Chuyển sang ký tự tiếp theo
j count_loop
```

end_count:

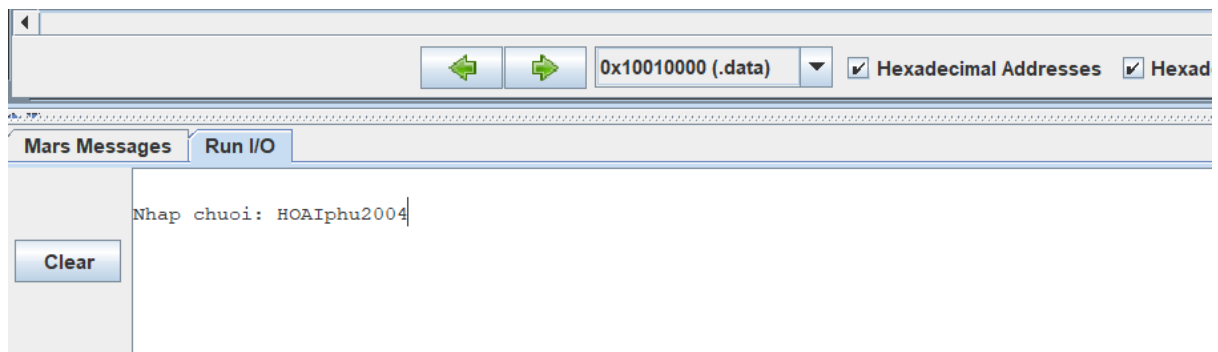
```
# In kết quả
li $v0, 4
la $a0, result_lowercase
syscall
li $v0, 1
move $a0, $t0
syscall
```

```
li $v0, 4
la $a0, result_uppercase
syscall
li $v0, 1
move $a0, $t1
syscall
```

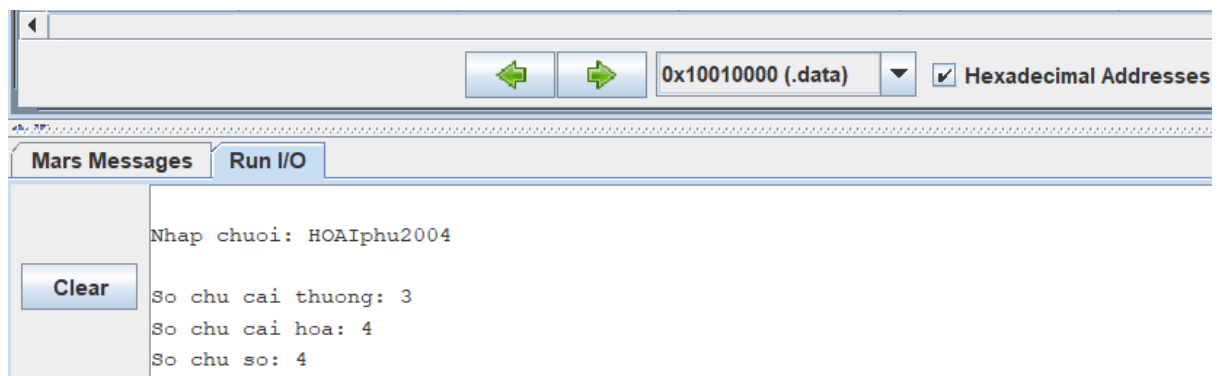
```
li $v0, 4
la $a0, result_digit
syscall
li $v0, 1
move $a0, $t2
syscall
```

#Chạy chương trình

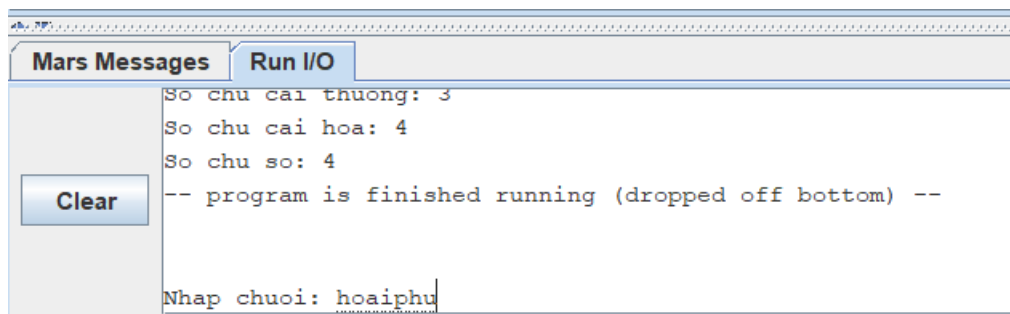
-Khi nhập HOAlphu2004



-Đáp án ra đúng



-Khi nhập hoaiphu



-Đáp án ra đúng

