

## LABORATORY 5

Họ tên: Vũ Quốc Bảo  
MSSV: 20225694

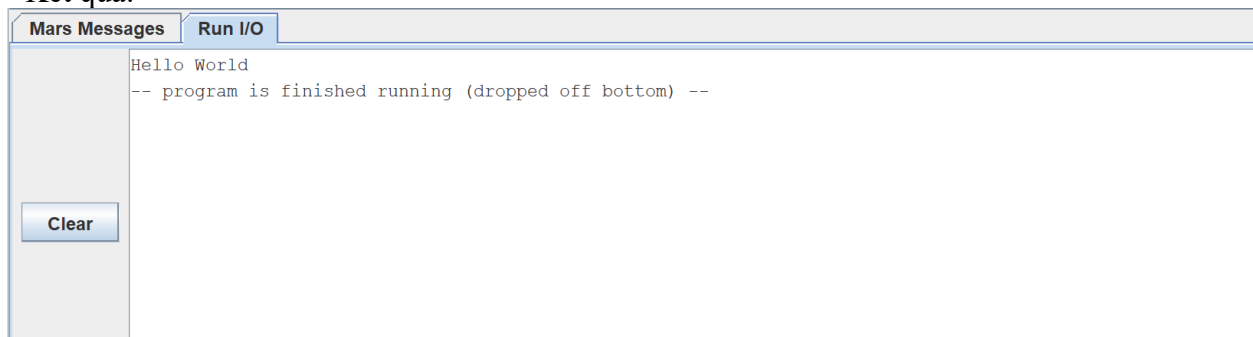
### Assignment 1:

```
1 .data
2 test: .asciiz "Hello World"
3 .text
4 li $v0, 4
5 la $a0, test
6 syscall
```

- Các bước:

- Khai báo chuỗi test là “Hello World” ở phần *.data*
- *li \$v0, 4*: gán giá trị v0 bằng 4 – print string
- *la \$a0, test*: gán giá trị của chuỗi test cho thanh ghi a0
- Dùng *syscall* để in chuỗi ra màn hình

- Kết quả:



### Assignment 2:

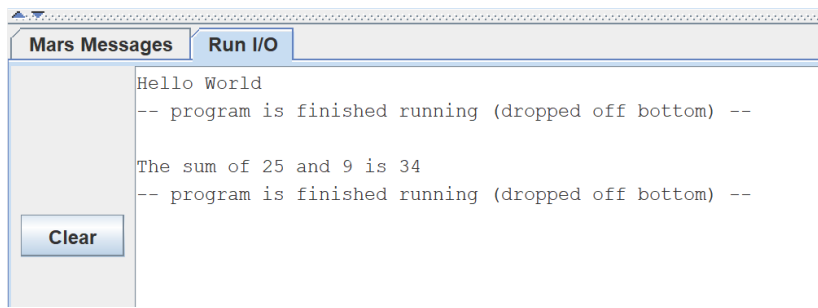
```

1  .data
2  message1: .ascii "The sum of "
3  message2: .ascii " and "
4  message3: .ascii " is "
5  .text
6  li $s0, 25
7  li $s1, 9
8  add $s2, $s0, $s1
9  li $v0, 4 # print string
10 la $a0, message1
11 syscall
12 li $v0, 1 # print decimal integer
13 add $a0, $zero, $s0
14 syscall
15 li $v0, 4
16 la $a0, message2
17 syscall
18 li $v0, 1
19 add $a0, $zero, $s1
20 syscall
21 li $v0, 4
22 la $a0, message3
23 syscall
24 li $v0, 1
25 add $a0, $zero, $s2
26 syscall

```

- Các bước:

- Trong mục **.data** khai báo các chuỗi "The sum of", " and ", " is "
- **li \$s0, 25**: gán giá trị cho s0 bằng 25
- **li \$s1, 9**: gán giá trị cho s1 bằng 9
- **add \$s2, \$s0, \$s1**: lưu giá trị s2 bằng tổng của 2 thanh ghi s0, s1
- Sử dụng xen kẽ cụm các lệnh **li \$v0, 4** và **li \$v0, 1** để in lần lượt ra màn hình message1, \$s0, message2, \$s1, message3, \$s2. Kết quả thu được như sau:



### Assignment 3:

```

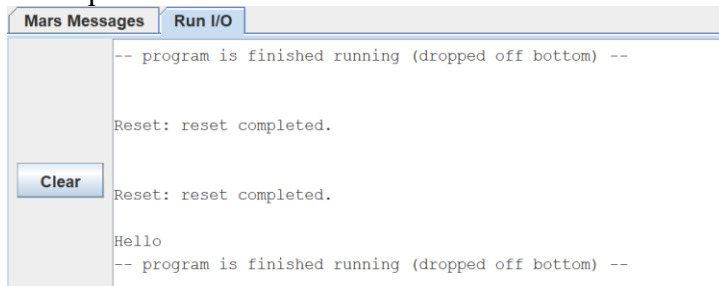
1  .data
2  x: .space 1000 # destination string x, empty
3  y: .asciiz "Hello" # source string y
4  .text
5  strcpy:
6  add $s0,$zero,$zero # s0 = i=0
7  L1:
8  la $a1, y
9  add $t1,$s0,$a1 # t1 = s0 + a1 = i + y[0]
10 # = address of y[i]
11 lb $t2,0($t1) # t2 = value at t1 = y[i]
12 la $a0, x
13 add $t3,$s0,$a0 # t3 = s0 + a0 = i + x[0]
14 # = address of x[i]
15 sb $t2,0($t3) # x[i]= t2 = y[i]
16 beq $t2,$zero,end_of_strcpy # if y[i]==0, exit
17 nop
18 addi $s0,$s0,1 # s0=s0 + 1 <-> i=i+1
19 j L1 # next character
20 nop
21 end_of_strcpy:
22 li $v0, 4
23 la $a0, x
24 syscall

```

- Các bước:

- Trong phần **.data** khai báo chuỗi rỗng x gồm 1000 phần tử, và chuỗi y có dạng “Hello”
- Phần **strcpy**: khai báo giá trị  $i = 0$
- Bắt đầu vòng lặp **L1**: trong mỗi vòng lặp, ta sẽ lấy 1 byte từ chuỗi nguồn y tại vị trí i rồi gán cho chuỗi đích x tại cùng 1 vị trí cho đến khi gặp kí tự NULL ( $y[i] = 0$ ) thì kết thúc vòng lặp và đưa kết quả của chuỗi x ra màn hình.

- Kết quả:



#### Assignment 4:

```

1  .data
2  string: .space 50
3  Message1: .asciiz "Nhap xau:"
4  Message2: .asciiz "Do dai la "
5  .text
6  main:
7  get_string:
8  li $v0, 54
9  la $a0, Message1
10 la $a1, string
11 la $a2, 50
12 syscall
13 get_length: la $a0, string # a0 = Address(string[0])
14 xor $v0, $zero, $zero # v0 = length = 0
15 xor $t0, $zero, $zero # t0 = i = 0
16 check_char: add $t1, $a0, $t0 # t1 = a0 + t0
17             # = Address(string[0]+i)
18 lb $t2, 0($t1) # t2 = string[i]
19 beq $t2, $zero, end_of_str # Is null char?
20 addi $v0, $v0, 1 # v0=v0+1->length=length+1
21 addi $t0, $t0, 1 # t0=t0+1->i = i + 1
22 j check_char
23 end_of_str:
24 end_of_get_length:
25 print_length:
26 sub $s1, $v0, 1
27 li $v0, 56
28 la $a0, Message2
29 move $a1, $s1
30 syscall

```

- Các bước:

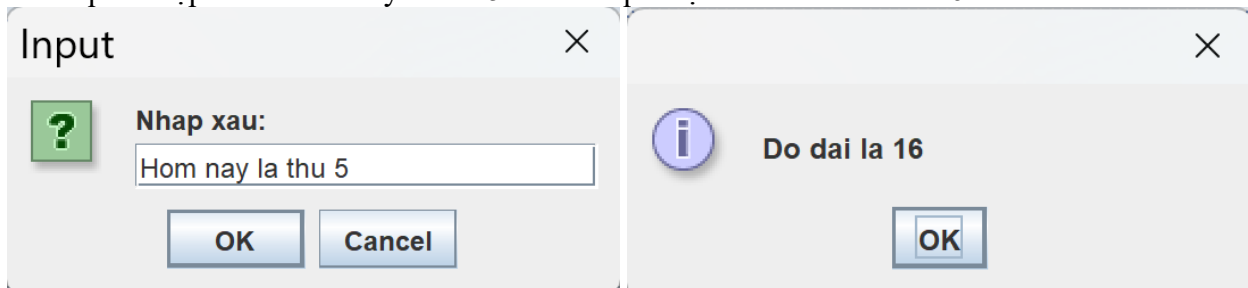
+ Phần **.data**:

- **.space 50**: Cấp phát 50 byte cho chuỗi.
- **Message1**: Khai báo một kết thúc chuỗi null là "Nhap xau:".
- **Message2**: Khai báo một kết thúc chuỗi null là "Do dai la ".

+ Phần **.text**:

- hàm **main** bắt đầu. Máy sẽ nhận chuỗi do người dùng nhập vào, sau đó tính toán độ dài của chuỗi và cuối cùng in ra màn hình độ dài của chuỗi đó.

- Kết quả: nhập xâu "Hom nay la thu 5" thì kết quả độ dài của xâu đó là 16.



### Assignment 5:

```

1  .data
2  string: .space 100
3  .text
4  input:
5      li $v0, 8
6      la $a0, string
7      li $a1, 21
8      syscall
9  get_length:
10     li $v0, 0 # length = 0
11     li $t0, 0 # i = 0
12 count_char:
13     add $t1, $a0, $t0 # địa chỉ của chuỗi string[i]
14     lb $t2, 0($t1)
15     beq $t2, $zero, exit1
16     addi $v0, $v0, 1
17     addi $t0, $t0, 1
18     j count_char
19 exit1:
20     addi $t0, $v0, -1 # t0 = length (không quan tâm đến '\n')
21     move $s0, $a0     # s0 = địa chỉ của chuỗi
22 print_reverse:
23     li $v0, 11
24     add $t1, $s0, $t0
25     lb $a0, 0($t1)
26     syscall
27     addi $t0, $t0, -1
28     blt $t0, $zero, exit2
29     j print_reverse
30 exit2:

```

## - Kết quả:

Mars Messages	Run I/O
	Reset: reset completed.
	Hello World!
Clear	!dlroW olleH -- program is finished running (dropped off bottom) --