

BÁO CÁO LAB02 MÔN THỰC HÀNH KIẾN TRÚC MÁY TÍNH (CE119-LAB03 / IT012-LAB05)

Câu 1: Thao tác với mảng

❖ Source code:

```
.data
array1:      .word 5, 6, 7, 8, 1, 2, 3, 9, 10, 4
size1:       .word 10
array2:      .byte 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
size2:       .word 16
array3:      .space 8
size3:       .word 8
s1: .asciiz " "
s2: .asciiz "\nARRAY1: "
s3: .asciiz "\nARRAY2: "
s4: .asciiz "\nARRAY3: "
s5: .asciiz "\nMang thu: "
s6: .asciiz "\nPhan tu thu: "
s7: .asciiz "INVALID."
.text
    la $s0, array1
    lw $a1, size1

    la $s1, array2
    lw $a2, size2

    la $s2, array3
    lw $a3, size3

# In array1
    add $t0, $0, $0
    li $v0, 4
    la $a0, s2
    syscall
loop:
    slt $t1, $t0, $a1
    beq $t1, $0, exit
    sll $t2, $t0, 2
    add $t3, $s0, $t2
    lw $t4, ($t3)
    li $v0, 1
    la $a0, ($t4)
    syscall
```

```

        li $v0, 4
        la $a0, s1
        syscall
        addi $t0, $t0, 1
        j loop
exit:

#In array2
        add $t0, $0, $0
        li $v0, 4
        la $a0, s3
        syscall
loop2:
        slt $t1, $t0, $a2
        beq $t1, $0, exit2
        add $t3, $s1, $t0
        lb $t4, ($t3)
        li $v0, 1
        la $a0, ($t4)
        syscall
        li $v0, 4
        la $a0, s1
        syscall
        addi $t0, $t0, 1
        j loop2
exit2:

#   array3[i] = array2[i] + array2[size2 - 1 - i]
        add $t0, $0, $0
loop3:
        slt $t1, $t0, $a3
        beq $t1, $0, exit3
#Tinh size2-1-i
        addi $t2, $a2, -1
        sub $t2, $t2, $t0
# array2[size2-1-i]
        add $t3, $s1, $t2
        lb $t4, ($t3)
#array2[i]
        add $t5, $s1, $t0
        lb $t6, ($t5)
#array2[i] + array2[size2-1-i]
        add $t7, $t6, $t4
#array3[i]
        add $t8, $s2, $t0

```

```
sb $t7, ($t8)
addi $t0, $t0, 1
j loop3
```

exit3:

#In array3

```
add $t0, $0, $0
li $v0, 4
la $a0, s4
syscall
```

loop4:

```
slt $t1, $t0, $a3
beq $t1, $0, exit4
add $t3, $s2, $t0
lb $t4, ($t3)
li $v0, 1
la $a0, ($t4)
syscall
li $v0, 4
la $a0, s1
syscall
addi $t0, $t0, 1
j loop4
```

exit4:

#Nhap mang thu may va phan tu nao

```
li $v0, 4
la $a0, s5
syscall
li $v0, 5
syscall
move $t0, $v0
```

```
sge $t1, $t0, 1
sle $t2, $t0, 3
bne $t1, $t2, EXIT
```

```
li $v0, 4
la $a0, s6
syscall
```

```
li $t1, 1
li $t2, 2
li $t3, 3
```

```
beq $t0, $t1, A_1
beq $t0, $t2, A_2
beq $t0, $t3, A_3
```

A_1:

```
li $v0, 5
syscall
move $t0, $v0

sge $t1, $t0, 0
slt $t2, $t0, $a1
bne $t1, $t2, EXIT

sll $t1, $t0, 2
add $t2, $s0, $t1
lw $t3, ($t2)
li $v0, 1
la $a0, ($t3)
syscall
j exit_pro
```

A_2:

```
li $v0, 5
syscall
move $t0, $v0

sge $t1, $t0, 0
slt $t2, $t0, $a2
bne $t1, $t2, EXIT

add $t2, $s1, $t0
lb $t3, ($t2)
li $v0, 1
la $a0, ($t3)
syscall
j exit_pro
```

A_3:

```
li $v0, 5
syscall
move $t0, $v0

sge $t1, $t0, 0
slt $t2, $t0, $a3
bne $t1, $t2, EXIT
```

```

add $t2, $s2, $t0
lb $t3, ($t2)
li $v0, 1
la $a0, ($t3)
syscall
j exit_pro

```

EXIT:

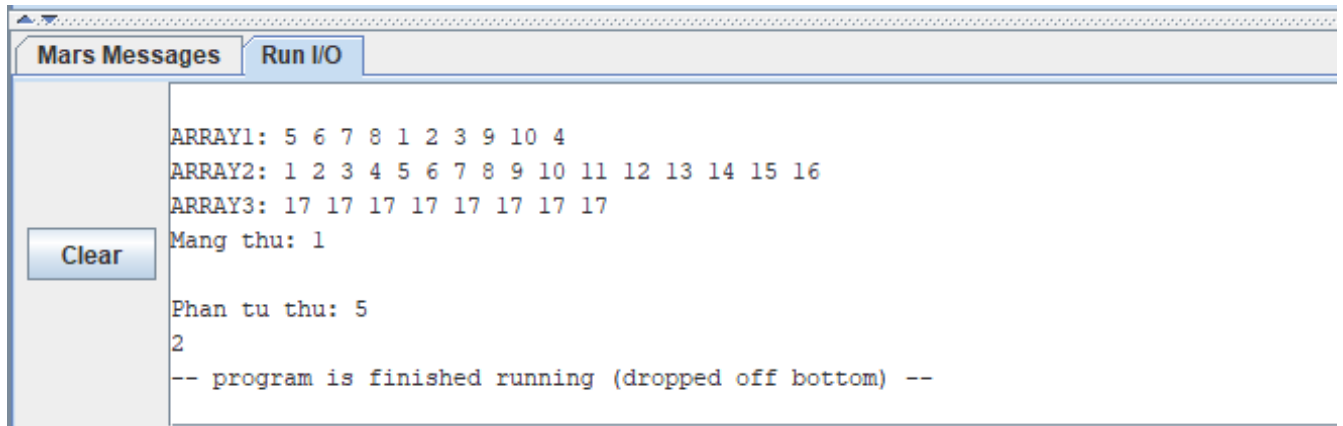
```

li $v0, 4
la $a0, s7
syscall

```

exit_pro:

❖ Kết quả:



Câu 2: Thao tác với con trỏ

❖ Source code:

```

.data
array1:      .word 5, 6, 7, 8, 1, 2, 3, 9, 10, 4
size1:       .word 10
array2:      .byte 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
size2:       .word 16
array3:      .space 8
size3:       .word 8
s1: .asciiz " "
s2: .asciiz "\nARRAY1: "
s3: .asciiz "\nARRAY2: "
s4: .asciiz "\nARRAY3: "
s5: .asciiz "\nMang thu: "
s6: .asciiz "\nPhan tu thu: "
s7: .asciiz "INVALID."

.text
la $s1, array1
lw $a1, size1

```

```
la $s2, array2
lw $a2, size2
la $s3, array3
lw $a3, size3
```

```
#In array1
add $t0, $0, $s1
sll $t1, $a1, 2
add $t2, $s1, $t1
li $v0, 4
la $a0, s2
syscall
```

```
loop1:
slt $t3, $t0, $t2
beq $t3, $0, exit1
lw $t4, ($t0)
li $v0, 1
la $a0, ($t4)
syscall
li $v0, 4
la $a0, s1
syscall
addi $t0, $t0, 4
j loop1
```

```
exit1:
```

```
#In array2
add $t0, $0, $s2
add $t2, $s2, $a2
li $v0, 4
la $a0, s3
syscall
```

```
loop2:
slt $t3, $t0, $t2
beq $t3, $0, exit2
lb $t4, ($t0)
li $v0, 1
la $a0, ($t4)
syscall
li $v0, 4
la $a0, s1
syscall
addi $t0, $t0, 1
j loop2
```

```
exit2:
```

```
#array3[i] = array2[i] + array2[size2 - 1 - i]
```

```
add $t0, $0, $s3
```

```
add $t1, $0, $s2
```

```
addi $t3, $a2, -1
```

```
add $t2, $t3, $s2 #array2[size2-1]
```

```
add $t3, $s3, $a3
```

```
loop3:
```

```
slt $t4, $t0, $t3
```

```
beq $t4, $0, exit3
```

```
lb $t5, ($t1)
```

```
lb $t6, ($t2)
```

```
add $t7, $t5, $t6
```

```
sb $t7, ($t0)
```

```
addi $t0, $t0, 1
```

```
addi $t1, $t1, 1
```

```
addi $t2, $t2, -1
```

```
j loop3
```

```
exit3:
```

```
#In array3
```

```
add $t0, $0, $s3
```

```
add $t2, $s3, $a3
```

```
li $v0, 4
```

```
la $a0, s4
```

```
syscall
```

```
loop4:
```

```
slt $t3, $t0, $t2
```

```
beq $t3, $0, exit4
```

```
lb $t4, ($t0)
```

```
li $v0, 1
```

```
la $a0, ($t4)
```

```
syscall
```

```
li $v0, 4
```

```
la $a0, s1
```

```
syscall
```

```
addi $t0, $t0, 1
```

```
j loop4
```

```
exit4:
```

```
#Nhap mang thu may va phan tu nao
```

```
li $v0, 4
```

```
la $a0, s5
```

```
syscall
li $v0, 5
syscall
move $t0, $v0
```

```
sge $t1, $t0, 1
sle $t2, $t0, 3
bne $t1, $t2, EXIT
```

```
li $v0, 4
la $a0, s6
syscall
```

```
li $t1, 1
li $t2, 2
li $t3, 3
```

```
beq $t0, $t1, A_1
beq $t0, $t2, A_2
beq $t0, $t3, A_3
```

A_1:

```
li $v0, 5
syscall
move $t0, $v0
```

```
sge $t1, $t0, 0
slt $t2, $t0, $a1
bne $t1, $t2, EXIT
```

```
sll $t1, $t0, 2
add $t1, $s1, $t1
lw $t2, ($t1)
li $v0, 1
la $a0, ($t2)
syscall
j exit_pro
```

A_2:

```
li $v0, 5
syscall
move $t0, $v0
```

```
sge $t1, $t0, 0
slt $t2, $t0, $a2
```



```
bne $t1, $t2, EXIT
```

```
add $t1, $s2, $t0
```

```
lb $t2, ($t1)
```

```
li $v0, 1
```

```
la $a0, ($t2)
```

```
syscall
```

```
j exit_pro
```

A_3:

```
li $v0, 5
```

```
syscall
```

```
move $t0, $v0
```

```
sge $t1, $t0, 0
```

```
slt $t2, $t0, $a3
```

```
bne $t1, $t2, EXIT
```

```
add $t1, $s3, $t0
```

```
lb $t2, ($t1)
```

```
li $v0, 1
```

```
la $a0, ($t2)
```

```
syscall
```

```
j exit_pro
```

EXIT:

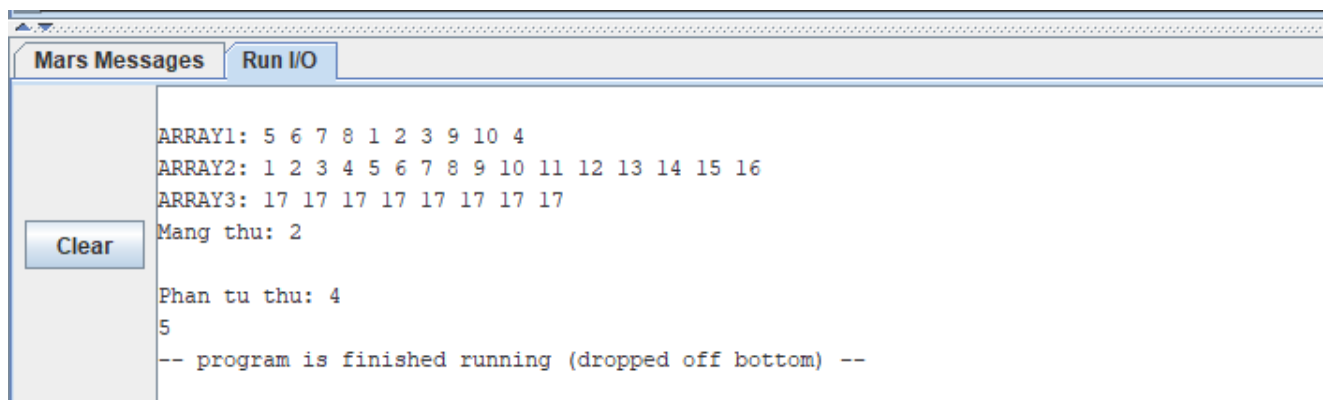
```
li $v0, 4
```

```
la $a0, s7
```

```
syscall
```

exit_pro:

❖ **Kết quả:**



The screenshot shows the Mars Messages window with the following output:

```
ARRAY1: 5 6 7 8 1 2 3 9 10 4
ARRAY2: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
ARRAY3: 17 17 17 17 17 17 17 17
Mang thu: 2

Phan tu thu: 4
5
-- program is finished running (dropped off bottom) --
```

Câu 3: (chỉ sử dụng con trỏ)

a) Nhập một mảng các số nguyên n phần tử (nhập vào số phần tử và giá trị của từng phần tử), xuất ra cửa sổ I/O của MARS theo từng yêu cầu sau:

- ✓ Xuất ra giá trị lớn nhất và nhỏ nhất của mảng
- ✓ Tổng tất cả các phần tử của mảng
- ✓ Người sử dụng nhập vào chỉ số của một phần tử nào đó và giá trị của phần tử đó được in ra cửa sổ

❖ Source code:

```
.data
array: .space 200
s1: .asciiz "\nNhap so luong phan tu: "
s2: .asciiz "\nNhap cac phan tu: "
s3: .asciiz " "
s4: .asciiz "\nMAX: "
s5: .asciiz "\nMIN: "
s6: .asciiz "\nSUM = "
s7: .asciiz "\nPhan tu thu: "
s8: .asciiz "INVALID."
.text

li $v0, 4
la $a0, s1
syscall

li $v0, 5
syscall
move $a1, $v0      # a1 : so luong phan tu

la $s1, array      #load add array len $s1
sll $t0, $a1, 2
add $t1, $s1, $t0 #Gioi han

add $t0, $0, $s1
loop:
    slt $t2, $t0, $t1
    beq $t2, $0, MIN_MAX
    li $v0, 5
    syscall
    sw $v0, ($t0)
    addi $t0, $t0, 4
    j loop

MIN_MAX:
    lw $s2, ($s1) #MAX
    lw $s3, ($s1) #MIN
    add $t0, $0, $s1
```

```
addi $t0, $t0, 4
```

```
loop_2:
```

```
    slt $t2, $t0, $t1  
    beq $t2, $0, print  
    lw $t3, ($t0)  
    slt $t4, $t3, $s3  
    bne $t4, $0, MIN  
    sgt $t4, $t3, $s2  
    bne $t4, $0, MAX  
    add $t0, $t0, 4  
    j loop_2
```

```
MIN:
```

```
    add $s3, $0, $t3  
    add $t0, $t0, 4  
    j loop_2
```

```
MAX:
```

```
    add $s2, $0, $t3  
    add $t0, $t0, 4  
    j loop_2
```

```
print:
```

```
    li $v0, 4  
    la $a0, s4  
    syscall  
    li $v0, 1  
    la $a0, ($s2)  
    syscall  
    li $v0, 4  
    la $a0, s5  
    syscall  
    li $v0, 1  
    la $a0, ($s3)  
    syscall
```

```
#Tong
```

```
    add $s0, $0, $0  
    add $t0, $0, $s1
```

```
loop_3:
```

```
    slt $t2, $t0, $t1  
    beq $t2, $0, print_sum  
    lw $t3, ($t0)  
    add $s0, $s0, $t3
```

```
addi $t0, $t0, 4
j loop_3
```

print_sum:

```
li $v0, 4
la $a0, s6
syscall
li $v0, 1
la $a0, ($s0)
syscall
```

#Nhap vao chi so va in ra

```
li $v0, 4
la $a0, s7
syscall
li $v0, 5
syscall
move $s0, $v0

sge $t2, $s0, 0
slt $t3, $s0, $a1
bne $t2, $t3, invalid
```

```
sll $t2, $s0, 2
add $t3, $s1, $t2
lw $t2, ($t3)
li $v0, 1
la $a0, ($t2)
syscall
j exit
```

invalid:

```
li $v0, 4
la $a0, s8
syscall
```

exit:

❖ **Kết quả:**

Mars Messages	Run I/O
<div>Clear</div>	Nhap so luong phan tu: 5
	1
	2
	3
	4
	5
	MAX: 5
	MIN: 1
	SUM = 15
	Phan tu thu: 3
4	
-- program is finished running (dropped off bottom) --	

b) **Nhập một mảng các số nguyên n phần tử (nhập vào số phần tử và giá trị của từng phần tử). Mảng này gọi là A.**

Chuyển dòng lệnh C dưới đây sang mã assembly của MIPS. Với các biến nguyên i, j được gán lần lượt vào thanh ghi $\$s0, \$s1$; và địa chỉ nền của mảng số nguyên A được lưu trong thanh ghi $\$s3$

*if ($i < j$) $A[i] = i$;
else $A[i] = j$;*

❖ **Source code:**

```
.data
a:      .word
s1:     .asciiz "\nNhap so luong phan tu: "
s2:     .asciiz "\nNhap cac phan tu: "
s3:     .asciiz " "
s4:     .asciiz "\nNhap i: "
s5:     .asciiz "\nNhap j: "
s6:     .asciiz "\nINVALID"
input_msg: .asciiz "(-1 to stop): "
output_msg: .asciiz "Array elements: "
```

```
.text
```

```
la $s2, a
li $v0, 4
la $a0, s2
syscall
```

```
li $a1, 0
add $t0, $s2, $0
```

```
loop_test:
li $v0, 4      # THONGBAO
la $a0, input_msg
```

syscall

li \$v0, 5

syscall

move \$t8, \$v0

beq \$t8, -1, EXIT

sw \$t8, 0(\$t0)

addi \$t0, \$t0, 4

addi \$a1, \$a1, 1

j loop_test

EXIT:

exit:

li \$v0, 4

la \$a0, s4

syscall

li \$v0, 5

syscall

move \$s0, \$v0

sge \$t1, \$s0, 0

slt \$t2, \$s0, \$a1

bne \$t1, \$t2, invalid

li \$v0, 4

la \$a0, s5

syscall

li \$v0, 5

syscall

move \$s1, \$v0

slt \$t0, \$s0, \$s1

sll \$t3, \$s0, 2

add \$t4, \$s2, \$t3

beq \$t0, \$0, else

sw \$s0, (\$t4)

j exit_p

else:

sw \$s1, (\$t4)

j exit_p

invalid:

```
li $v0, 4
la $a0, s6
syscall
j exit
```

exit_p:

```
#In array1
li $v0, 4
la $a0, output_msg
syscall
```

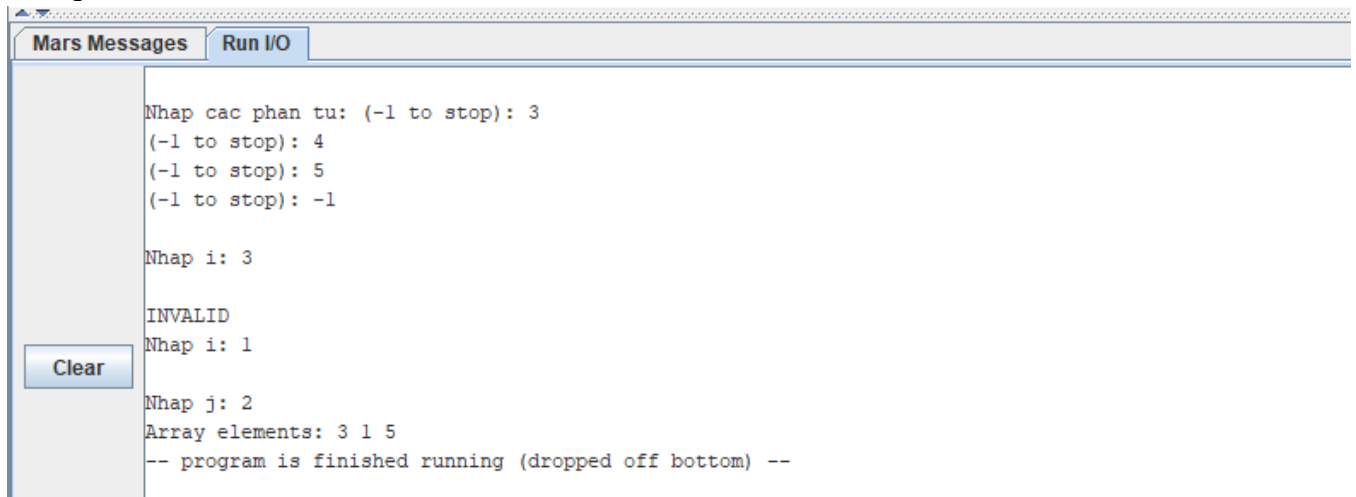
```
add $t0, $0, $s2
sll $t1, $a1, 2
add $t2, $s2, $t1
```

loop1:

```
slt $t3, $t0, $t2
beq $t3, $0, exit1
lw $t4, ($t0)
li $v0, 1
la $a0, ($t4)
syscall
li $v0, 4
la $a0, s3
syscall
addi $t0, $t0, 4
j loop1
```

exit1:

❖ Kết quả:



```
Mars Messages Run I/O

Nhap cac phan tu: (-1 to stop): 3
(-1 to stop): 4
(-1 to stop): 5
(-1 to stop): -1

Nhap i: 3

INVALID
Nhap i: 1

Nhap j: 2
Array elements: 3 1 5
-- program is finished running (dropped off bottom) --
```