

BÁO CÁO THỰC HÀNH LAB 05

Họ và tên: *Nguyễn Đỗ Quang* - MSSV: *20520720*

THỰC HÀNH

2.a. In ra tất cả các phần tử của mảng array1 và array2:

```
.data
newline: .ascii "\n"
space: .ascii " "
error: .ascii "Khong the truy cap"

.text
#in mảng array1
la $s0, array1      #lưu địa chỉ array1 vào $s0
la $s1, 40($s0)      #lưu địa chỉ của ô nhớ sau phần tử cuối của mảng vào $s1
move $t0, $s0        #dùng $t0 làm con trỏ, trỏ đến địa chỉ $s0

loop:
lw $a0, ($t0)        #đọc giá trị con trỏ $t0 trỏ đến
li $v0, 1            #lệnh syscall 1 để xuất
syscall

li $v0, 4
la $a0, space
syscall

addi $t0, $t0, 4      #trỏ đến phần tử tiếp theo
bne $t0, $s1, loop    #lặp cho đến phần tử cuối thì kết thúc lặp

#in mảng array2
la $s2, array2        #lưu địa chỉ array2 vào $s2
la $s3, 16($s2)        #lưu địa chỉ của ô nhớ sau phần tử cuối của mảng vào $s3
move $t0, $s2         #dùng $t0 làm con trỏ, trỏ đến địa chỉ $s0
loop1:
lb $a0, ($t0)
li $v0, 1
syscall
li $v0, 4
la $a0, space
syscall
addi $t0, $t0, 4
bne $t0, $s3, loop1
```

bytes thay cho word (16 bytes)
tương tự như các bước xuất ở array1 (lb thay cho lw)

2.b. Gán giá trị cho mảng array3

la \$s4,array3	#lưu địa chỉ array3 vào \$s4
la \$s5,32(\$s4)	#lưu địa chỉ của ô nhớ sau phần tử cuối của mảng vào \$s5
move \$t0,\$s4	#\$t0 là con trỏ, trỏ đến phần tử đầu của array3
move \$t1,\$s2	#\$t1 là con trỏ, trỏ đến phần tử đầu của array2
subi \$t2,\$s3,4	#\$t2 là con trỏ, trỏ đến phần tử cuối của array2
loop2:	
lb \$t3,(\$t1)	#đọc giá trị con trỏ \$t1 trỏ đến
lb \$t4,(\$t2)	#đọc giá trị con trỏ \$t2 trỏ đến
add \$t5,\$t3,\$t4	#array2[i] + array2[size2 - 1 - i]
sb \$t5,(\$t0)	#array3[i] = array2[i] + array2[size2 - 1 - i]
addi \$t0,\$t0,4	#trỏ đến phần tử tiếp theo
addi \$t1,\$t1,4	#trỏ đến phần tử tiếp theo
subi \$t2,\$t2,4	#trỏ đến phần tử phía trước
bne \$t0,\$s5,loop2	#lặp cho đến phần tử cuối thì kết thúc lặp

2.c. Nhập vào thứ tự mảng và chỉ số phần tử, sau đó xuất ra phần tử tương ứng

li \$v0,5	#nhập 1 hoặc 2 hoặc 3 để chọn mảng cần tìm
syscall	
bgt \$v0,3,error1	# Không nhập đúng thì lỗi
ble \$v0,0,error1	
beq \$v0,1,assign1	#nếu nhập vô 1 thì nhảy đến nhãn assign1
bne \$v0,1,arr2	
assign1:	
move \$t0,\$s0	
j next1	
arr2:	
beq \$v0,2,assign2	#nếu nhập vô 2 thì nhảy đến nhãn assign2
bne \$v0,2,arr3	
assign2:	
move \$t0,\$s2	
j next2	
arr3:	
move \$t0,\$s4	
j next2	

next1:	# next1 để xử lý đọc dữ liệu ở arr1 (do kiểu dữ liệu word)
li \$v0,5	
syscall	
bge \$v0,10,error1	
sll \$t1,\$v0,2	#offset của index so với phần tử đầu
add \$t2,\$t1,\$t0	
lw \$a0,(\$t2)	# load dữ liệu lên thanh ghi
li \$v0,1	
syscall	#In ra phần tử cần tìm
j exit	
next2:	# next2 để xử lý đọc dữ liệu ở arr2,arr3 (do kiểu dữ liệu byte)
li \$v0,5	
syscall	
bge \$v0,16,error1	
move \$t1,\$v0	#offset của index so với phần tử đầu
add \$t2,\$t1,\$t0	
lb \$a0,(\$t2)	# load dữ liệu lên thanh ghi
li \$v0,1	
syscall	# In ra phần tử cần tìm
j exit	
error1:	
li \$v0,4	
la \$a0,error	# Lỗi in ra error
syscall	
exit:	

BÀI TẬP

3.a Nhập mảng số nguyên n phần tử

#Hàm nhập n phần tử và giá trị của từng phần tử	
.data	
.text	
#nhap n phan tu	
li \$v0,5	#lệnh syscall 5 để nhập n phần tử
syscall	
move \$t2,\$v0	#Luu n vào trong \$t2
move \$a0,\$t2	#luu n vào trong a0 để chuẩn bị cấp phát động
sll \$a0,\$a0,2	#nhân 4 để ra số bytes cần cấp phát
li \$v0,9	#sử dụng lệnh syscall 9(sbrk) để cấp phát bộ nhớ trên heap
syscall	
move \$t0,\$v0	#\$t0 là địa chỉ của mảng
sll \$t3,\$t2,2	#\$t3 là độ dài
add \$t3,\$t0,\$t3	#địa chỉ sau phần tử cuối của mảng lưu vào \$t3
add \$t1,\$t0,\$0	#con trỏ \$t0 trỏ đến phần tử đầu của mảng
#nhap tung phan tu	
loop: li \$v0,5	#lệnh syscall 5 để nhập từng phần tử trong loop
syscall	
sw \$v0,(\$t1)	#luu giá trị mới nhập vào ô nhớ \$t1 trỏ tới
add \$t1,\$t1,4	#\$t1 trỏ tới phần tử tiếp theo
bne \$t3,\$t1,loop	#dừng khi đến phần tử cuối

a.1 Xuất ra giá trị lớn nhất và nhỏ nhất

Giá trị lớn nhất

lw \$s2,(\$t0)	#đọc giá trị đầu tiên của mảng gán làm max(max=A[0])
add \$t1,\$t0,4	#\$t1 trỏ đến phần tử thứ 2(A[1])
loop2:	
lw \$s3,(\$t1)	#đọc giá trị từ địa chỉ của A[i]
bgt \$s3,\$s2,assign	#nếu giá trị A[i]>max nhảy đến nhãn assign
resume:	
addi \$t1,\$t1,4	#nhảy đến phần tử tiếp theo
bne \$t3,\$t1,loop2	#nếu chưa đến phần tử cuối thì tiếp tục lặp
beq \$t3,\$t1,exit	#đến phần tử cuối thì nhảy đến nhãn exit kết thúc vòng lặp
assign:	
move \$s2,\$s3	#gán max=A[i]
j resume	#nhảy về nhãn resume để tiếp tục vòng lặp
exit:	
#xuất Max:	

```

move $a0,$s2
li $v0,1
syscall

```

Giá trị nhỏ nhất: (Tương tự như max dòng 5 thay vì bgt thì đổi thành blt)

```

lw $s2,($t0)      #đọc giá trị đầu tiên của mảng gán làm max(min=A[0])
add $t1,$t0,4      #t1 trỏ đến phần tử thứ 2(A[1])
loop2:
lw $s3,($t1)       #đọc giá trị từ địa chỉ của A[i]
blt $s3,$s2,assign  #nếu giá trị A[i]<min nhảy đến nhãn assign
resume:
addi $t1,$t1,4      #nhảy đến phần tử tiếp theo
bne $t3,$t1,loop2   #nếu chưa đến phần tử cuối thì tiếp tục lặp
beq $t3,$t1,exit     #đến phần tử cuối thì nhảy đến nhãn exit kết thúc vòng lặp
assign:
move $s2,$s3        #gán min=A[i]
j resume            #nhảy về nhãn resume để tiếp tục vòng lặp
exit:
#xuất Min:
move $a0,$s2
li $v0,1
syscall

```

3.a.2 Nhập mảng số nguyên n phần tử, xuất ra tổng các phần tử trong mảng

```

#Tính Tổng
add $t1,$t0,$0      #địa chỉ mảng lưu vào $t1
add $s0,$s0,$0      #$s0 chứa tổng các phần tử
#Cộng các giá trị lại:
loop1:
lw $s1,($t1)        #đọc giá trị phần tử A[i]
add $s0,$s0,$s1      #S+=A[i]
addi $t1,$t1,4       #nhảy đến giá trị tiếp theo
bne $t3,$t1,loop1    #tiếp tục lặp khi chưa đến phần tử cuối
#Xuất tổng:
move $a0,$s0
li $v0,1
syscall

```

3.a.3 Nhập mảng số nguyên n phần tử và nhập chỉ số phần tử, sau đó xuất ra giá trị của phần tử

```
#Nhập index
li $v0,5
syscall
move $t4,$v0      #$t4 là index A[i]
sll $t4,$t4,2      #$t4 là độ dời
add $t4,$t4,$t0     #$t4 là địa chỉ của phần tử cần tìm
Xuất phần tử cần tìm:
lw $a0,($t4)
li $v0,1
syscall
```

3.b. Nhập mảng số nguyên n phần tử, sau đó biên dịch chương trình

C	MIPS
if(i < j) A[i] = i; else A[i] = j;	.data .text #nhập n phần tử li \$v0,5 syscall move \$t2,\$v0 #\$t2 là size của mảng move \$a0,\$t2 sll \$a0,\$a0,2 } đổi ra bytes li \$v0,9 #lệnh cấp phát địa chỉ ô nhớ trên heap syscall move \$s3,\$v0 #lưu địa chỉ mảng được cấp phát vào \$s3 sll \$t3,\$t2,2 #\$t3 là độ dời add \$t3,\$s3,\$t3 #địa chỉ phần tử cuối lưu vào \$t3 add \$t1,\$s3,\$0 #\$t1 trở đến phần tử đầu của mảng #nhập từng phần tử loop: li \$v0,5 syscall sw \$v0,(\$t1) #lưu giá trị với nhập vào ô nhớ \$t1 trở đến add \$t1,\$t1,4 #phần tử tiếp theo bne \$t3,\$t1,loop #đến cuối thì dừng vòng lặp li \$s0,2 #VD:gán i=1 li \$s1,1 #VD:gán j=2 move \$t4,\$s0 move \$t5,\$s1 sll \$t4,\$t4,2 #độ dời của i sll \$t5,\$t5,2 #độ dời của j

	add \$t4,\$t4,\$s3 #địa chỉ A[i] bgt \$s0,\$s1,assign #i>j nhảy đến assign sw \$s0,(\$t4) #A[i] = i; j exit #nhảy đến exit assign:sw \$s1,(\$t4) # A[i] = j; exit:
--	--