Kiểm tra giữa kì- Kiến trúc máy tính-130938

Tên: Trần Minh Quân
Mssv: 20205015
Mã lớp: 130938

Bài 1:

Source code:

```
.data
mss1:		.asciiz		"Nhap so nguyen duong N: "
mss2:		.asciiz		"Chu so lon nhat: "
space:		.asciiz		" "
commas:		.asciiz		", "

.text
main:
li  $v0, 51	#Nhap so
la  $a0, mss1
syscall
add $s0, $a0,0
li $s1, 10

la $a0, mss2  #Ket qua
li $v0, 4
syscall

li $s3, 0	#Khoi tao s3=0
loop:
beq $s0, 0, end	#Thuc hien chia s0 cho 10 den khi bang 0
div $s0, $s1
mflo $s0
mfhi $s2
slt $s4, $s3, $s2	#Luu so du lon nhat
beq $s4, 1, swap
j loop
swap:
add $s3, $s2, 0
```

j loop

end:

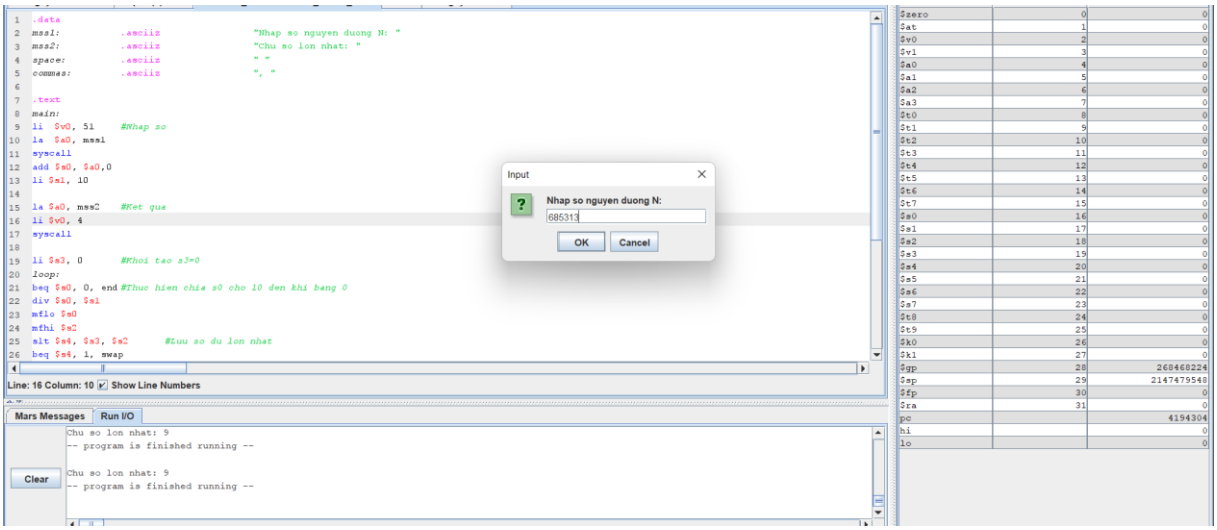
li $v0, 1

add $a0, $s3, 0

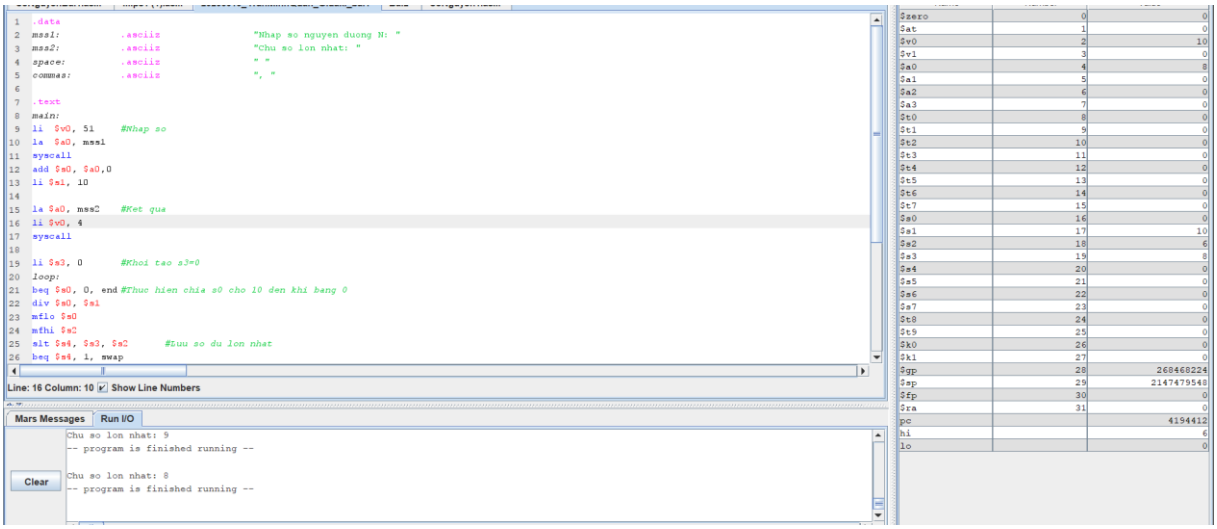
syscall

li $v0, 10

syscall



Kết quả:



Giải thích:

Thực hiện chia liên tiếp số đã cho($s0) với 10, mỗi lần chia so sánh số dư($s2) với biến $s3 hiện lưu, nếu lớn hơn thì đẩy vào $s3 => được số lớn nhất. Kết thúc quá trình khi $s0=0

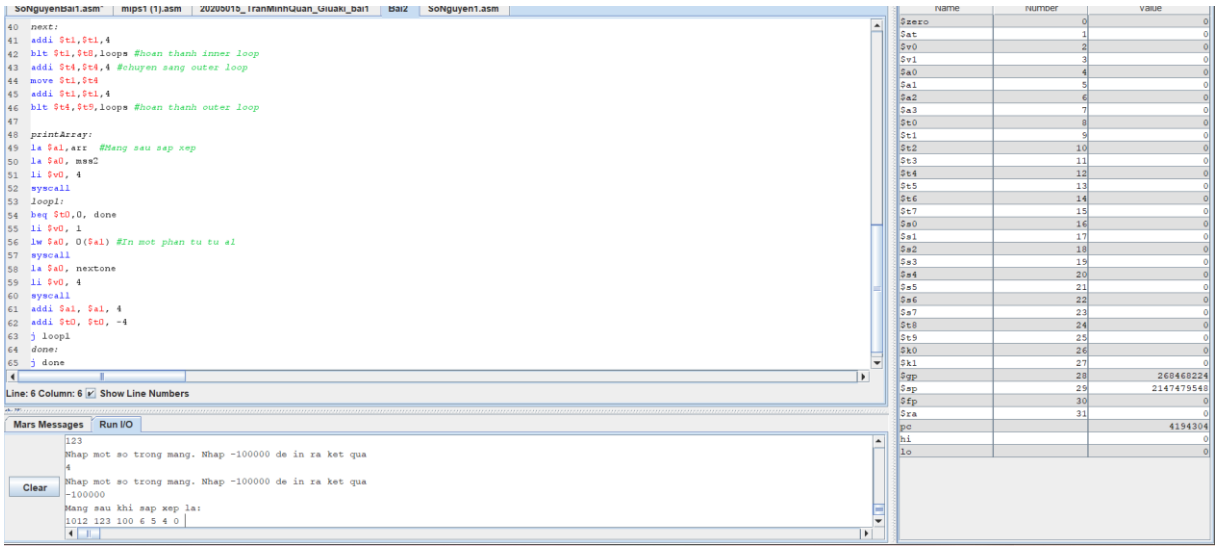
Bài 2:

Source code:

```
.data
arr: .space 5000
mss1: .asciiz "Nhap mot so trong mang. Nhap -100000 de in ra ket qua\n"
mss2: .asciiz "Mang sau khi sap xep la:\n"
nextone: .asciiz " "
.text
main:
la $a1, arr # Khoi tao mang
li $t0, 0
li $t1,-100000 #Key ket thuc
loop:

la $a0, mss1
li $v0, 4
syscall
li $v0, 5
syscall
beq $v0,$t1,sort #Nhap xong chuyen sang sort
addi $t0,$t0,4
sw $v0, ($a1)
addi $a1, $a1, 4 # tang so ptu len 1
j loop

sort:

la $t4, arr #Khoi tao outer loop
la $t1, arr #So tuong ung trong inner loop
addi $t1,$t1,4 #Bat dau tu ptu so 2
la $t8,arr
add $t8,$t0,$t8
la $t9,arr
add $t9,$t0,$t9
addi $t9,$t9,-4
loops:
lw $t2,($t4) #so trong inner tuong ung
lw $t3,($t1) #so trong outer
```

```
bgt $t2,$t3,next #Doi cho neu t2>t3
sw $t3,($t4) #swap 2 so
sw $t2,($t1)
next:
addi $t1,$t1,4
blt $t1,$t8,loops #hoan thanh 1 inner loop
addi $t4,$t4,4 #chuyen sang outer loop
move $t1,$t4
addi $t1,$t1,4
blt $t4,$t9,loops #hoan thanh outer loop

printArray:
la $a1,arr  #Mang sau sap xep
la $a0, mss2
li $v0, 4
syscall
loop1:
beq $t0,0, done
li $v0, 1
lw $a0, 0($a1) #In mot phan tu tu mang a1
syscall
la $a0, nextone
li $v0, 4
syscall
addi $a1, $a1, 4
addi $t0, $t0, -4 #Dung lai khi het ptu, t0=0
j loop1
done:
li $v0, 10
syscall
```

Kết quả



Giải thích:

-In liên tục đến khi gặp -100000 thì ra kết quả(-100000 đơn giản đóng vai trò key để dừng, có thể thay bằng bất kì thứ khác)

-Sắp xếp bằng cách tạo 2 vòng lặp (Độ phức tạp o(n^2)). Với mỗi số trong outer loop($t3), so tất cả các số trong inner loop($t2) với nó. Nếu nhỏ hơn thì đổi chỗ 2 số, lớn hơn giữ nguyên

-Cứ như vậy được dãy sắp xếp

Bài 3: