

ASSIGNMENT 1: Số nguyên

Bài 5: Nhập số nguyên dương N có từ 2 chữ số trở lên. In ra màn hình các chữ số của N theo chiều ngược lại.

Bài làm

I. Phân tích thực nghiệm

- Tạo ra mảng gồm 10 phần tử
- Ở mỗi vòng lặp thì ta sẽ chia cho 10 để lấy số dư
- Sau đó đã sẽ lưu các giá trị dư vào 1 mảng rồi khởi tạo số mới đảo ngược
- Với mỗi vòng lặp ta in ra tổng các số cộng lại hoàn thành để thành số đảo ngược

```
.data
A: .space 10          #tao mang co kich thuoc la 10
output: .asciiz "So dao nguoc la: "

.text
li      $v0,5          #Nhap so
syscall

la      $a0, A          #luu dia chi mang A vao a0
add     $s0, $0, $v0    #Luu so vua nhap vao thanh ghi $s0
addi    $t0, $0, 10     #dat so chia la 10
add     $a1, $0, 0      #i=0

while:
beq     $s0, $0, done   #vong lap while (i!=0)
div     $s0, $s0, $t0    #chia s0 cho 10 moi vong lap
mfhi    $t3              #t3=s0 % 10
sll     $t2, $a1,2       #t2=i*4
add     $t4, $a0, $t2    #A[i]= A[0]+i*4
sw      $t3,0($t4)       #load A[i]=t3
addi    $a1, $a1,1       #i++
```

```

j      while      #vong lap
done:
add    $a1, $a1, -1    #i=i-1
add    $t2, $0, 0      #j=0
daonguocso:
beq    $a1, $0, end    #vong lap while i!=0
add    $t1, $a1, $0    #t1=i
addi   $t0, $0, 1      #t0=1
Khoitao:
beq    $t1, $0, next   #while(t1!=0)
mul    $t0, $t0, 10     #t0*=10
add    $t1, $t1, -1     #i--
j      Khoitao
next:
sll    $t3, $t2, 2      #t3=j*4
add    $t9, $a0, $t3    #t9 la dia chi cua A[j]
lw     $t8, 0($t9)      #t8=A[j]
mul    $t0, $t0, $t8    #t0=t0*A[j]
add    $s0, $s0, $t0    #tinh tong
add    $a1, $a1, -1     # i--
addi   $t2, $t2, 1      #j++
j      daonguocso
end:
lw     $t8, 4($t9)
add    $s0, $s0, $t8    #tong= tong+A[j]
print:
li     $v0, 56
la     $a0, output
add    $a1, $s0, $0
syscall

```

Đoạn code của chương trình

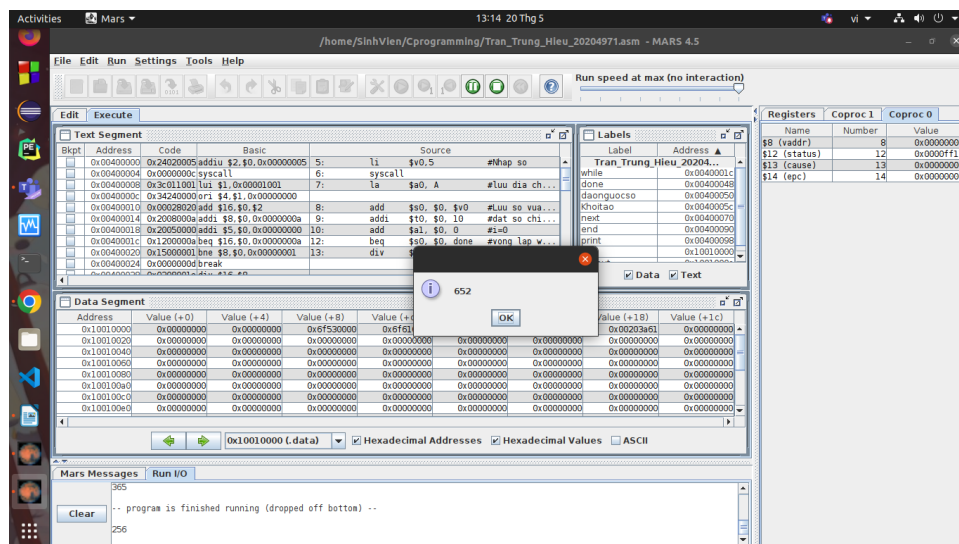
	Tran_Trung_Hieu_20204971.asm	mips2.asm
1	.data	
2	A: .space 10	#tao mang co kich thuoc la 10
3	output: .asciiz "So dao nguoc la: "	
4	.text	
5	li \$v0,5	#Nhap so
6	syscall	
7	la \$a0, A	#luu dia chi mang A vao a0
8	add \$s0, \$0, \$v0	#Luu so vua nhap vao thanh ghi \$s0
9	addi \$t0, \$0, 10	#dat so chia la 10
10	add \$a1, \$0, 0	#i=0
11	while:	
12	beq \$s0, \$0, done	#vong lap while (i!=0)
13	div \$s0, \$s0, \$t0	#chia s0 cho 10 moi vong lap
14	mfhi \$t3	#t3=s0 % 10
15	sll \$t2, \$a1, 2	#t2=i*4
16	add \$t4, \$a0, \$t2	#A[i]= A[0]+i*4
17	sw \$t3, 0(\$t4)	#load A[i]=t3
18	addi \$a1, \$a1, 1	#i++
19	j while	#vong lap
20	done:	
21	add \$a1, \$a1, -1	#i=i-1
22	add \$t2, \$0, 0	#j=0
23	daonguocso:	
24	beq \$a1, \$0, end	#vong lap while i!=0
25	add \$t1, \$a1, \$0	#t1=i
26	addi \$t0, \$0, 1	#t0=1
27	Khoitao:	
28	beq \$t1, \$0, next	#while(t1!=0)
29	mul \$t0, \$t0, 10	#t0*=10
30	add \$t1, \$t1, -1	#i--
31	j Khoitao	
32	next:	
33	sll \$t3, \$t2, 2	#t3=j*4
34	add \$t9, \$a0, \$t3	#t9 la dia chi cua A[j]
35	lw \$t8, 0(\$t9)	#t8=A[j]
36	mul \$t0, \$t0, \$t8	#t0=t0*A[j]
37	add \$s0, \$s0, \$t0	#tinh tong
38	add \$a1, \$a1, -1	# i--
39	addi \$t2, \$t2, 1	#j++
40	j daonguocso	
41	end:	
42	lw \$t8, 4(\$t9)	
43	add \$s0, \$s0, \$t8	#tong= tong+A[j]

```

44 print:
45     li      $v0, 56
46     la      $a0, output
47     add     $a1, $s0, $0
48     syscall
49
50

```

- Output ra màn hình giá trị đã đảo ngược



Kết quả sau khi chạy chương trình với giá trị nhập vào là 256

ASSIGNMENT 2: Mảng

Bài 2: Nhập mảng số nguyên từ bàn phím. In ra màn hình cặp phần tử liền kề có tổng nhỏ nhất.

I) Phân tích thực nghiệm

- Nhập số lượng phần tử
- Nhập giá trị của từng phần tử của mảng rồi lưu vào ngăn xếp
- Tạo giá trị min-so-far thực hiện cộng 2 phần tử liên tiếp trong mảng rồi gán vào t2. Nếu tổng 2 phần tử nhỏ hơn min-so-far thì gán vào \$t2
- Làm lần lượt từ đầu tới cuối các phần tử của mảng

II) Đoạn code của chương trình

```

.data
mes1: .ascii "Nhap so phan tu cua mang"
mes2: .ascii "Nhap phan tu vao mang"
mes3: .ascii "cap phan tu can tim la"
mes4: .ascii " ,"

.text
addi $a2,$zero, 999999999999

NhapN:
li      $v0, 51
la      $a0, mes1
syscall          #dung syscall de nhap so n
addi    $s0, $a0,0      # gan gia tri N vao $s0

Nhapmang:
li      $v0, 51
la      $a0,mes2
syscall

addi    $s1,$a0,0
addi    $fp,$fp,-4
sw      $s1,0($fp)
subi    $s0,$s0,1
beqz    $s0,summin
j       Nhapmang

summin:
beq     $zero,$fp,print
lw      $t0,0($fp)
addi    $fp,$fp,4
beq     $zero,$fp,print
lw      $t1,0($fp)
add     $t2,$t0,$t1 # tong 2 phan tu lien tiep

```

```

slt          $t4,$t2,$a2
bnez        $t4, min
addi        $t0,$zero,0 # reser lai gia tri o t0
addi        $t1,$zero,0 #reset lai gia tri o t1
beq         $zero,$fp,print
j           summin
min:
add         $a2,$t2,$zero
add         $t8,$t0,$zero
add         $t9,$t1,$zero
addi        $t0,$zero,0
addi        $t1,$zero,0
j           summin
print:

li          $v0, 4
la          $a0, mes3          # in ra cau B
syscall

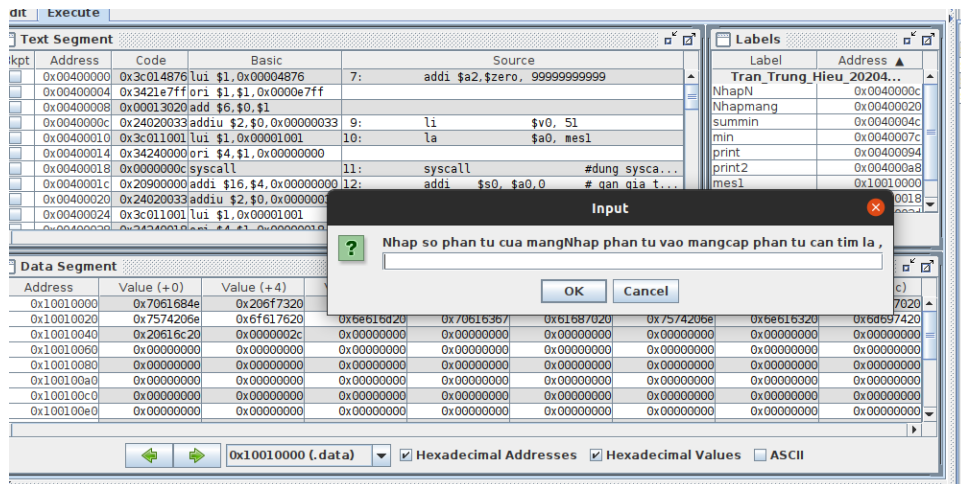
j           print2
print2:
li          $v0,1
add         $a0,$t8,$zero
syscall

li          $v0,4
la          $a0,mes4
syscall

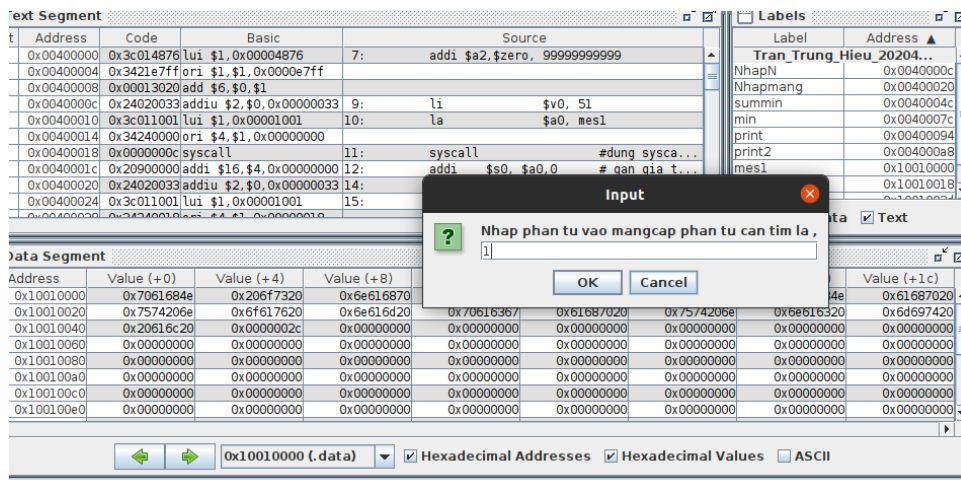
li          $v0,1
add         $a0,$t9,$zero
Syscall

```

III) Kết quả thực nghiệm



Nhập số lượng phần tử trong mảng



Nhập lần lượt các phần tử vào mảng

```
-- program is finished running (dropped off bottom)
cap phan tu can tim la ,2 ,1
-- program is finished running (dropped off bottom)
```

Kết quả của chương trình khi mảng gồm 5 phần tử là {1,2,3,4,5}

ASSIGNMENT 3: String

Bài 5:

.data

Message: .ascii "string:"

string: .space 100

```

.text
li $v0, 54                                #input string
la $a0, Message
la $a1, string
la $a2, 100
syscall
la $a0,string                            #$a0=address[string(0)]
addi $t0,$a0,-1
addi $t1,$a0,100                          #$a1=address[string(0)]+length max string input
li $t3,91
li $t4,32
li $t5,12
loop: # tinh address[string(max)]
addi $t1,$t1,0                            #$t1 = $t1 + $0
lb $v0,0($t1)                             # nap du lieu tu bo nho vao thanh ghi $a1
slti $at,$v0,12                           #v0<12 ->at=1 nguoc lai at=0
beq $at,$0,swap                           #at=0->jump swap
addi $t1,$t1,-1                            #$t1=$t1-1
j loop
swap:
addi $t0,$t0,1                            #$t0=$t0+1
lb $a1,0($t0)                             #nap du lieu tu bo nho vao thanh ghi $a1
beq $a1,$t4,swap                           #$a1=$t4-< jump swap
slt $at,$t3,$a1
bne $at,$0,endswap                        #a1>a3 jump endswap
addi $a1,$a1,32                            #a1=a1+32
j endswap                                #jump end swap
endswap:
sb $a1,0($t0)                             #nap du lieu tu thanh ghi vao lai bo nho
slt $at,$t0,$t1
beq $at,$0,endmain                        #$t0=$t1 jump endmain

```


j swap

endmain:

li \$v0,10

#exit

syscall

Đoạn code của chương trình

```
loop: # tính address[str.  
      addi $t1,$t1,0  
      lb $v0,0($t1)  
      slti $at,$v0,12  
      beq $at,$0,swap  
      addi $t1,$t1,-1  
      j loop
```

- \$t1 là địa chỉ của phần tử tối đa của string khai báo
- \$v0 chứa bit mà địa \$t1 đang chứa
- Chương trình chạy với địa chỉ giảm dần kết thúc khi gặp ký tự null
- Kết thúc \$t1 chứa địa chỉ của ký tự null

```
swap:  
      addi $t0,$t0,1  
      lb $a1,0($t0)  
      beq $a1,$t4,swap  
      slt $at,$t3,$a1  
      bne $at,$0,endswap  
      addi $a1,$a1,32  
      j endswap
```

- \$t0 chứa địa chỉ của phần tử con trỏ đang trỏ đến trong mảng
- Nạp dữ liệu từng bit của mảng vào thanh ghi nếu là chữ thường kết thúc, chữ hoa thì đổi thành chữ thường bằng cách thay đổi mã ascii đang chứa trong thanh ghi

```

endswap:
    sb $a1,0($t0)
    slt $at,$t0,$t1
    beq $at,$0,endumain
    j swap

```

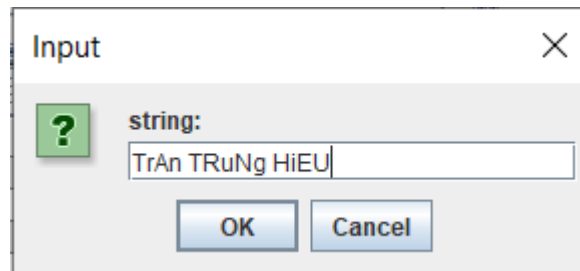
- Nạp lại dữ liệu đã thay đổi từ thanh ghi vào bộ nhớ vòng lặp kết thúc khi đến địa chỉ cuối cùng của mảng

```

endumain:
    li $v0,10
    syscall

```

Thoát chương trình



Input

string: TrAn TRuNg HiEU

OK Cancel

n	a	r	t	u	r	t	h	g	n	\n	u	e	i	\0	\0	\0	\0		
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0