

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
BÁO CÁO MÔN HỌC
Tên học phần : Thực hành Kiến trúc máy tính IT3280
Học kỳ 20212 năm học 2021-2022

ĐỀ A6,B2,C5

TOPIC A

Assignment 6

Code:

.data

text: .asciiz "Nhap so n = "

.text

main:

la \$a0, text

jal print_text #in text

jal read_int #doc so nhap vao

move \$s1, \$v0 # \$s1 = \$v0

li \$s2, 0 # i =0

li \$s3, 0 # j =0

div:

ble \$s1, 0, next

div \$s1, \$s1,8

mfhi \$a0 #luu so du

jal stack_push

addi \$s2, \$s2, 1 #i++

j div

next: #in lan luot tung so du trong stack

bge \$s3, \$s2, print_done

jal stack_pop

addi \$a0, \$v0, 0 #luu gia tri vua duoc pop

jal print_int

addi \$s3, \$s3, 1

j next

print_done:

j end_program

print_text:

li \$v0, 4

syscall

jr \$ra

read_int:

li \$v0, 5

syscall

jr \$ra

print_int:

li \$v0, 1 #print interger

syscall

jr \$ra

print_space:

addi \$a0,\$zero, 32 # \$a0 = " "

li \$v0, 11 #print character

```
syscall
```

```
jr $ra
```

stack_push:

```
addi $sp, $sp, -4 # $sp -=4
```

```
sw $a0, 0($sp) #put a word to stack
```

```
jr $ra
```

stack_pop:

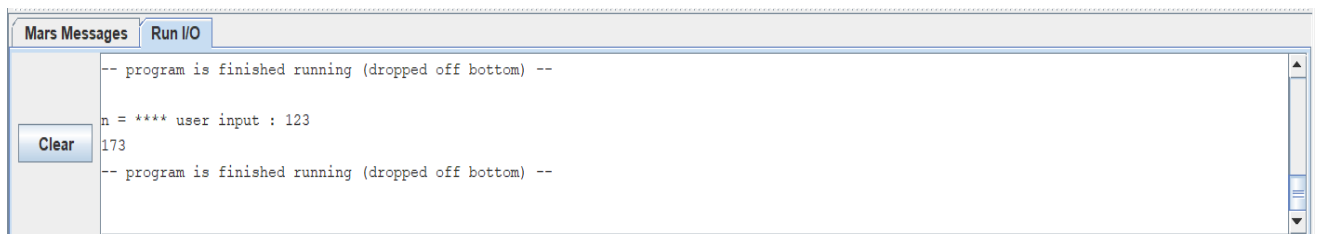
```
lw $v0, 0($sp) #get a word form stack
```

```
addi $sp, $sp, 4 # $sp +=4
```

```
jr $ra
```

end_program:

Kết quả:



Cách thực hiện:

- Sử dụng div để tìm số dư của số n khi chia cho 8
- Tiếp theo ta lưu số dư này vào stack
- Lấy từ stack ra và in từng số dư

TOPIC B

Assignment 2:

Code:

```
.data
```

NhapN: .asciiiz "Nhap so phan tu mang: "

Xuat_Mang: .asciiiz "Mang vua nhap la: \n"

Xuat: .asciiiz "Nhap phan tu thu "

Xuat_Min: .asciiiz "Cap phan tu lien ke co tong nho nhat la : "

Space: .asciiiz " va "

tong: .word 0

a: .word 0:100

.text

.globl main

main:

la \$a0, NhapN # in NhapN

li \$v0, 4

syscall

li \$v0, 5 # nhap n va luu vao \$s0

syscall

move \$s0, \$v0

li \$t0, 0 # khoi tao vong lap , i = 0

la \$a1, a #load a vao \$a1

li \$s1, 0 # khoi tao tong 2 phan tu min =0

Enter:

```
blt $t0, $s0, enter_Element    # Neu i < n thi enter_Element
```

```
j out
```

enter_Element:

```
la $a0, Xuat    # print string Xuat
```

```
li $v0, 4
```

```
syscall
```

```
li $v0, 1        # print index
```

```
move $a0, $t0
```

```
syscall
```

```
li $v0, 5        # nhap so nguyen va luu vao mang
```

```
syscall
```

```
sw $v0, ($a1)    #luu phan tu moi nhap vao mang
```

```
lw $v1,-4($a1)   # load phan tu lien truoc phan tu moi nhap
```

```
addi $t0, $t0, 1    #Tang chi so
```

```
addi $a1, $a1, 4
```

```
# $s3 la bien lien truoc bien nhap vao hien tai,
```

```
# $s2 la tong hien tai
```

```
# $s1 la tong be nhat hien tai,
```

```
# $s4,$s5 luu gia tri 2 phan tu co tog be nhat
```

```
add $s2,$v0,$s3    # tinh lai tong moi
```

```
add $s3,$v0,$zero # cap nhat lai $s3
```

neu so phan tu la 1 thi nhap tiep

addi \$t3,\$zero,1 # neu so phan tu la 1 thi nhap tiep

slt \$t1,\$t3,\$t0 # 1 < t0 ? 1:0

beq \$t1,\$zero,Enter

#neu so phan tu la 2 thi gan \$s1 = tong 2 so do

addi \$t3,\$zero,2 #neu so phan tu la 2 thi gan \$s1 = tong 2 so do

slt \$t1,\$t3,\$t0 # 2 < t0 ? 1:0

beq \$t1,\$zero,FirstSum

so sanh voi tong min

slt \$t2,\$s2,\$s1 # \$t2 = \$s2 < \$s1 (tong moi < tong cu ? 1: 0

beq \$t2,\$zero,Enter # t2=0 (tong moi > tong cu => quay lai enter

add \$s1,\$s2,\$zero # t2=1 (tong moi < tong cu => gan lai \$s1)

#gan lan luot 2 phan tu do

add \$s5,\$v0,\$zero

add \$s4,\$v1,\$zero

j Enter

FirstSum:

add \$s1,\$s2,\$zero

add \$s5,\$v0,\$zero

add \$s4,\$v1,\$zero

j Enter

out:

```
la $a0,Xuat_Min      #print string Xuat_Min
```

```
li $v0,4
```

```
syscall
```

```
li $v0, 1      # print 2 phan tu min
```

```
move $a0, $s4
```

```
syscall
```

```
la $a0,Space      #print string Space
```

```
li $v0,4
```

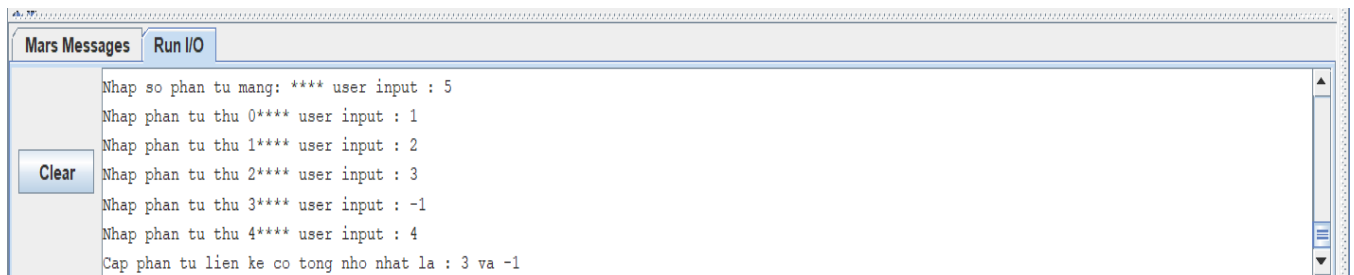
```
syscall
```

```
li $v0, 1      # print 2 phan tu min
```

```
move $a0, $s5
```

```
syscall
```

- Kết quả:



Cách thực hiện:

- Sử dụng system call để nhập vào từng phần tử trong mảng. Tiếp theo ta check trường hợp chỉ có 1 phần tử thì quay lại nhập tiếp ; trường hợp có 2 phần tử thì ta gọi đến FirstSum để tính tổng lúc đầu tiên.
- Sau đó ta sử dụng lệnh slt để check xem tổng mới có nhỏ hơn tổng cũ hay không. Nếu tổng mới lớn hơn thì quay lại lệnh Enter nhập tiếp, nếu tổng mới nhỏ hơn thì cập nhật lại tổng cũ = tổng mới và lưu lại 2 giá trị 2 biến liền kề lần lượt vào 2 thanh ghi \$s4,\$s5.

TOPIC C

Assignment 5

Code

```
.data
String: .space 100
Nhap: .asciiz "Nhap vao 1 chuoi: "
Xuat: .asciiz "Chuoi vua nhap chuyen qua chu thuong la: "
.text
.globl main
main:

addi $v0,$0,4
la $a0,Nhap
syscall

la $a0,String
addi $a1,$0,100
addi $v0,$0,8
syscall
#chuoi vua nhap dang nam trong String

la $s0,String
add $t0,$0,$0
add $t1,$s0,$t0
lb $s1,0($t1)
# $s0 la dia chi o dau tien cua chuoi
# $t0 la bien chay
# $t1 la dia chi cua tung ki tu cua chuoi
# $s1 la gia tri cua tung ki tu cua chuoi
```


Next:

add \$t1,\$s0,\$t0

lb \$s1,0(\$t1)

#2 dòng trên để cập nhật \$t1 và \$s1

bne \$s1,\$0, VongLap #\$s1 bằng 0 là hết chuỗi

addi \$v0,\$0,4

la \$a0,Xuat

syscall

addi \$v0,\$0,4

la \$a0,String

syscall

addi \$v0,\$0,10

syscall

VongLap:

li \$t6,10

beq \$s1,\$t6,BayQuaKhongKiemTra

li \$t4,'a'

slt \$t3,\$s1,\$t4

bne \$t3,\$0,VongLapKhongPhaiKiTuThuong #ki tu nay nho hon a nen ko phai la chu thuong

li \$t4,'z'

slt \$t3,\$t4,\$s1

bne \$t3,\$0,VongLapKhongPhaiKiTuThuong #ki tu nay lon hon Z nen ko phai la chu thuong

BayQuaKhongKiemTra:

#Neu khong phai la 2 truong hop tren thi ki tu la thuong

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

```
j Next
```

VongLapKhongPhaiKiTuThuong:

```
li $t4, ''
```

```
beq $s1,$t4,KhoangTrang          #if ki tu la space thi khong cong them 32
```

```
addi $s5,$s1,+32
```

```
sb $s5,0($t1)
```

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

```
j Next
```

KhoangTrang:

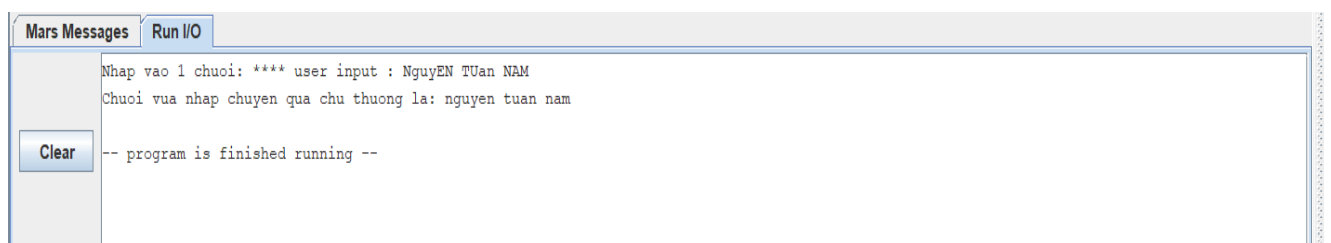
```
addi $s5,$s1,+0
```

```
sb $s5,0($t1)
```

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

```
j Next
```

Kết quả:



Cách thực hiện:

- Ta sử dụng label VongLap: để check
- Nếu kí tự nhập vào nhỏ hơn kí tự 'a' và lớn hơn kí tự 'z' thì sẽ chui vào nhãn VongLapKhongPhaiKiTuThuong:
- Trong VongLapKhongPhaiKiTuThuong:, ta cộng thêm 32 cho kí tự chữ hoa trở thành chữ thường, ngoài ra ta check thêm label KhoangTrang: để vẫn giữ nguyên là kí tự Space