

Bài thực hành số 5

Lớp: 147799 – Học phần: Thực Hành Kiến Trúc Máy Tính

Họ và tên: Lương Văn Khanh - MSSV: 20225728

Asg1: (A1)

- Code :

.data

mes1:.asciiz "Nhap so nguyen duong N"

mes0:.asciiz "Day la so am, nhap lai \n"

mes2:.asciiz "Cac so thoa man la : "

mes3:.asciiz " "

.text

main:

li \$v0,51 #nhap N

la \$a0,mes1

syscall

move \$t1,\$a0

checkN:

bltz \$t1,lai

li \$v0,4 # in ra mes2

la \$a0,mes2

syscall

li \$t8,3

li \$t9,5

```
li $s0,0 #gan i=0
```

```
j loop
```

```
lai:
```

```
li $v0,4
```

```
la $a0,mes0
```

```
syscall
```

```
j main # Nhap lai N
```

```
loop:
```

```
slt $t0,$s0,$t1
```

```
beq $t0,$0,end # if i>=N => end
```

```
rem $t2,$s0,$t8
```

```
beqz $t2,print # if i chia het cho 3
```

```
rem $t3,$s0,$t9
```

```
beqz $t3,print # if i chia het cho 5
```

```
j con
```

```
print:
```

```
li $v0,1 #in so i thoa man
```

```
move $a0,$s0
```

```
syscall
```

```
li $v0,4
```

```
la $a0,mes3 #in dau cach
```

```
syscall
```

```
j con
```

```
con:
```

```
addi $s0,$s0,1 #i++
```

j loop

end:

li \$v0,10

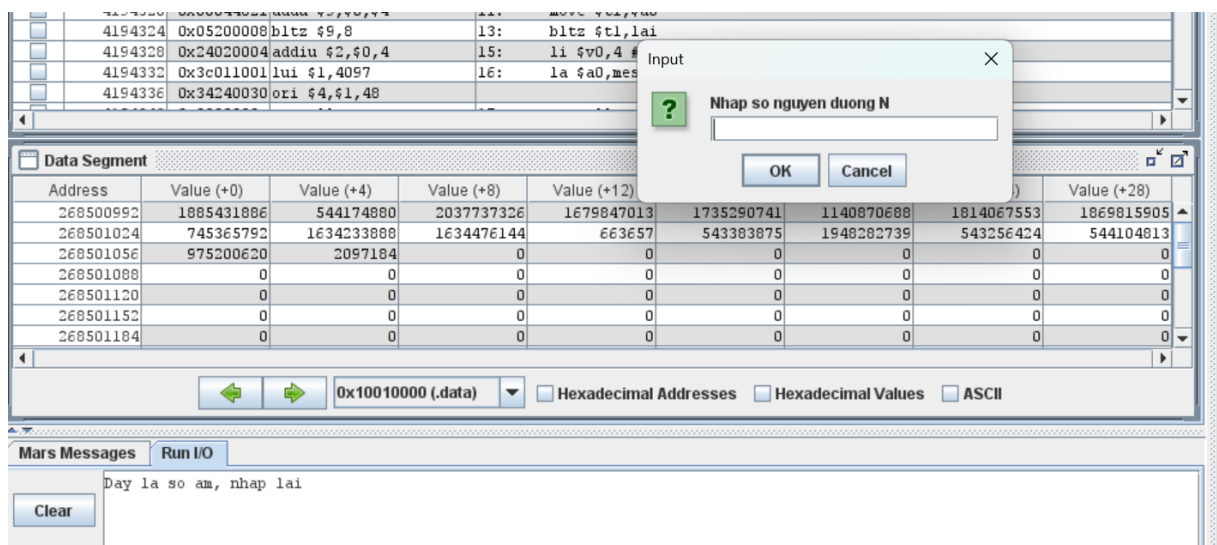
syscall

- Giải thích code : Đây là chương trình in ra những số chia hết cho 3 hoặc 5 nhỏ hơn N.

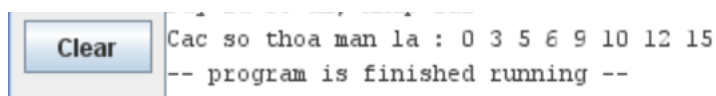
Ban đầu ta nhập N từ bàn phím. Nếu $N < 0$ thì hiện lỗi ra cửa sổ RunIO và yêu cầu nhập lại.

- Kết quả thực hiện:

- Nếu nhập $N = -1$ thì báo lỗi , yêu cầu nhập lại



- Nếu nhập $N = 17$



Asg2: (B10)

- Code :

.data

arr: .space 100

in: .asciiz "Nhap vao kich thuc cua mang: "

error1: .asciiz "Mang khong ton tai so le\n"

error2: .asciiz "Mang khong ton tai so chan\n"

error3: .asciiz "Khong ton tai phan tu thoa man\n"

out: .asciiz "Phan tu thoa man la: "

.text

input:

li \$v0,51

la \$a0,in

syscall

move \$v0,\$a0

ble \$v0, 0, input #Kiem tra do dai cua mang

addi \$s1, \$0, 0 #max1 = 0 (So le lon nhat)

addi \$s2, \$0, 1 #min2 = 1 (So chan be nhat)

add \$s0, \$0, \$v0 #n : length of array

la \$s3, arr #array

#Nhap cac phan tu cua mang, tim phan tu chan be nhat va gan vao min2,

tim ra 1 phan tu le va gan vao max1

addi \$t1, \$0, 0 #i = 0

for1: bge \$t1, \$s0, end_for1

sll \$t2,\$t1,2 # t2 = 4*i

add \$t0, \$s3, \$t2 #val = arr+i

li \$v0,5

syscall

sw \$v0, 0(\$t0) #*val = input

addi \$v1, \$0, 2

div \$v0, \$v1

```

mfhi          $v1

if1:  bne      $v1, $0, else1  #neu so du khac 0 thi nhay den else1

    #if(*val % 2 == 0) do

        beq      $s2, 1, set_min_2          #Neu gap so chan dau tien thi gan min2
la so chan do

        bge      $v0, $s2, end_if_else1 #v0 >= s2 thi nhay den end_if_else1

        #if(min2 == 1 || *val < min2) do set_min_2

set_min_2:

        add      $s2, $0, $v0  #min2 = *val

        j        end_if_else1

else1: beq      $s1, $0, set_min_1          #them vao

        bge      $v0, $s1, end_if_else1    #v0 >= s1 thi nhay den end_if_else1

        #if(max1 == 0 || *val < max1) do set_max_1

set_min_1:

        add      $s1, $0, $v0  #max1 = *val

end_if_else1:

        addi     $t1, $t1, 1

        j        for1

end_for1:

        beq      $s1, 0, invalid1          #Neu gia tri cua max1 khong thay doi => khong
ton tai so le trong mang

        beq      $s2, 1, invalid2          #Neu gia tri cua min2 khong thay doi => khong
ton tai so chan trong mang

        addi     $t1, $0, 0                #i = 0

for2:  bge      $t1, $s0, end_for2

        add      $t2, $t1, $t1

        add      $t2, $t2, $t2

```

```

        add      $t0, $s3, $t2    #val = arr+i

        lw       $v0, 0($t0)     #*val = input

        addi     $v1, $0, 2

        div      $v0, $v1

        mfhi     $v1

if2:    beq      $v1, $0, end_if2

        bge      $v0, $s2, end_if2

        bge      $s1, $v0, end_if2

        #if(*val % 2 == 1 && *val < min2 && *val > max1) do

        add      $s1, $0, $v0

end_if2:

        addi     $t1, $t1, 1

        j        for2

end_for2:

        bge      $s1, $s2, invalid3    #Neu max1 >= min2    => khong ton tai phan tu
le thoa man dieu kien

        j        success                #In ra phan tu thoa man


invalid1:

        addi     $v0, $0, 4

        la       $a0, error1

        syscall

        j        end

invalid2:

        addi     $v0, $0, 4

        la       $a0, error2

```

```
syscall
```

```
j      end
```

invalid3:

```
addi    $v0, $0, 4
```

```
la       $a0, error3
```

```
syscall
```

```
j      end
```

success:

```
addi    $v0, $0, 4
```

```
la       $a0, out
```

```
syscall
```

```
addi    $v0, $0, 1
```

```
add     $a0, $0, $s1
```

```
syscall
```

```
j      end
```

end:

```
addi    $v0, $0, 10
```

```
syscall
```

- Giải thích code: Chương trình in ra phần tử lẻ lớn nhất nhỏ hơn mọi số chẵn trong mảng.

Hàm for1 tìm số chẵn bé nhất(nếu có) gán vào min2(thanh ghi s2)

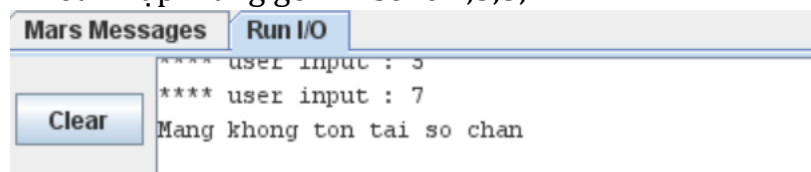
Hàm for2 tìm số lẻ lớn nhất(nếu có) gán vào max1(thanh ghi s1)

Hàm end_for2 so sánh max1 và min2

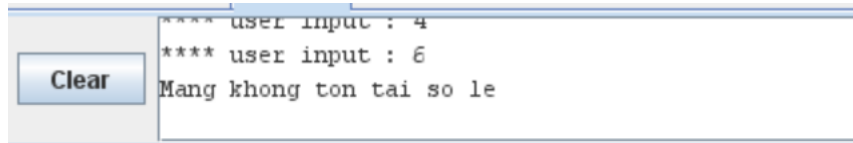
- Kết quả thực hiện:

- Nếu nhập kích thước mảng âm thì yêu cầu nhập lại

- Nếu nhập mảng gồm 4 số lẻ 1,3,5,7



- Nếu nhập mảng gồm 3 số chẵn 2,4,6

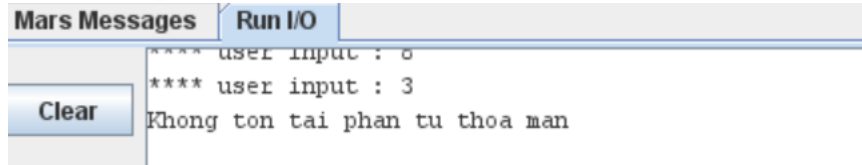


```

**** user input : 4
**** user input : 6
Mang không tồn tại số lẻ

```

- Nếu nhập mảng gồm 5 số : 2,4,6,8,3

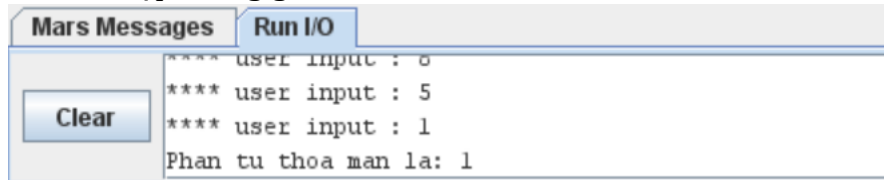


```

**** user input : 8
**** user input : 3
Không tồn tại phần tử thỏa mãn

```

- Nếu nhập mảng gồm 6 số : 2,4,6,8,5,1



```

**** user input : 8
**** user input : 5
**** user input : 1
Phần tử thỏa mãn là: 1

```

Asg3: (C7)

- Code:

.data

xau: .space 100 # Khai báo mảng

in: .asciiz "Nhập xau ki tu: "

out: .asciiz "Xau ki tu sau khi chuyen: "

.text

main:

li \$v0,54

la \$a0,in

la \$a1,xau

la \$a2,100

syscall

Gọi hàm để chuyển đổi xâu kí tự

la \$a0, xau # truyền địa chỉ của xâu kí tự vào \$a0

jal chuyen_doi

In ra xâu kết quả

li \$v0, 4

la \$a0, out # load địa chỉ của chuỗi cần in

syscall

In xâu kết quả

li \$v0, 4

la \$a0, xau # load địa chỉ của xâu kết quả

syscall

Kết thúc chương trình

li \$v0, 10

syscall

chuyen_doi:

 # Lặp qua từng ký tự trong xâu kí tự

 la \$t0, 0 # chỉ số của ký tự

loop:

 lb \$t1, 0(\$a0) # đọc ký tự tại địa chỉ \$a0 và lưu vào \$t1

beqz \$t1, end # nếu ký tự là NULL, thoát khỏi vòng lặp

Kiểm tra ký tự là chữ thường

li \$t2, 97 # ASCII của 'a'

li \$t3, 122 # ASCII của 'z'

blt \$t1, \$t2, NOT # nếu ký tự nhỏ hơn 'a', không phải chữ thường

bgt \$t1, \$t3, NOT # nếu ký tự lớn hơn 'z', không phải chữ thường

thường => hoa

addi \$t1, \$t1, -32

sb \$t1, 0(\$a0)

j NEXT

NOT:

li \$t2, 65 # ASCII của 'A'

li \$t3, 90 # ASCII của 'Z'

blt \$t1, \$t2, NEXT # nếu ký tự nhỏ hơn 'A', không phải chữ hoa

bgt \$t1, \$t3, NEXT # nếu ký tự lớn hơn 'Z', không phải chữ hoa

hoa => thường

add \$t1, \$t1, 32

sb \$t1, 0(\$a0)

NEXT:

```

addi $a0, $a0, 1    # di chuyển địa chỉ để trỏ vào ký tự tiếp theo trong chuỗi ký
tự

j loop

```

end:

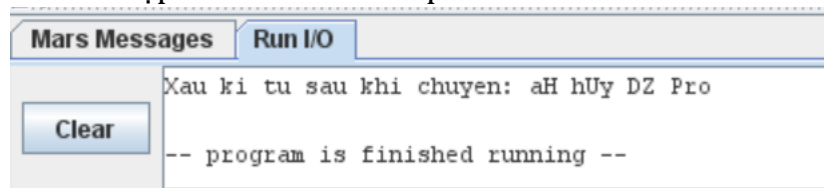
```
jr $ra
```

- Giải thích code : Đây là chương trình nhập vào 1 chuỗi ký tự, chuyển các ký tự thường thành hoa, hoa thành thường và in ra màn hình. Duyệt từng ký tự của chuỗi, nếu ký tự là chữ thường (từ 'a' đến 'z') thì - 32, nếu ký tự là chữ hoa (từ 'A' đến 'Z') thì + 32, chạy đến khi gặp ký tự NULL thì thoát khỏi vòng lặp.

Những ký tự không phải chữ cái sẽ được giữ nguyên.

- Kết quả thực hiện:

- Nếu nhập chuỗi " Ah HuY dz pRO "



- Nếu nhập chuỗi " aH huY dZ cO 102 ! "

