

Báo cáo kiểm tra giữa kỳ 2022.2

Thực hành kiến trúc máy tính

Phạm Thành Lập-20215076

Assigment 19.A

- Code

```
Error: .asciiz "So nhap vao phai lon hon 0!\n"

.text
main:
    li $v0, 4
    la $a0, Message
    syscall                                #in ra thong bao nhap so
    li $v0, 5
    syscall                                #input number
    move $s0, $v0                          #truyen gia tri vua nhap vao N
    bltz $s0, PrintError                    #in ra loi neu so ko dung dang
    j Prime

exit:
    li $v0, 10
    syscall                                #exit

Prime:
    addi $s0, $s0, 1                        # N+1
    li $t0, 2                              # i = 2

IsPrime:
    bge $t0, $s0, PrintResult              # if i = N in ra ket qua
    div $s0, $t0                           # chia N cho i
    mfhi $t1                               # lay du luu vao $t1
    beqz $t1, Prime                        # if du = 0 return Prime
    addi $t0, $t0, 1                       # i++
    j IsPrime

PrintResult:
    li $v0, 4
    la $a0, Message1
    syscall                                #in ra thong bao ket qua
    li $v0, 1
    addi $a0, $s0, 0
    syscall
    j exit

PrintError:
    li $v0, 4
    la $a0, Error
    syscall
    j main
```

- Result

```
-- program is finished running --

Nhap vao 1 so nguyen duong N: 15
So nguyen to nho nhat lon hon N la: 17
-- program is finished running --
```

- *Phân tích cách thực hiện:*

+ Sau khi nhập vào số nguyên dương N ta tăng N lên 1 đơn vị sau đó thiết lập $i = 2$ thực hiện chia N cho i và lấy dư nếu khi i tăng đến N thì tất cả các phép chia dư đều khác 0 thì ta in ra N sẽ là số dương nhỏ nhất lớn hơn giá trị ta nhập vào, còn ngược lại nếu khi chia ra dư bằng 0 thì ta sẽ quay lại tăng N lên sau đó đặt lại i bằng 2 và tiếp tục thực hiện bước trên

Assigment 6.B

- Code

```

.data
A: .word 0:100
Message: .asciiiz "Nhap so phan tu mang: "
Message1: .asciiiz "Nhap so: "
Message2: .asciiiz "Day so am giam ma khong lam thay doi thu tu so duong la:\n"
Error: .asciiiz "So phan tu mang phai lon hon bang 2!\n"
Space: .asciiiz " "

.text
main:
    li $v0, 4
    la $a0, Message
    syscall                                #in ra thong bao nhap so phan tu day
    li $v0, 5
    syscall                                # input number
    move $s0, $v0                          # truyen gia tri vua nhap vao N
    blt $s0, 2, PrintError                  #in ra thong bao loi neu so phan tu khong dung
    la $a1, A                               # load adress of A to a1
    j input_array

exit:
    li $v0, 10
    syscall                                #exit

input_array:
    beq $t2, $s0, CheckArray                # if i = N
    add $t0, $a1, $t1                       # t0 = address of A + $t1 (A[i])
    li $v0, 4
    la $a0, Message1
    syscall                                #nhap vao chuoi
    li $v0, 5
    syscall                                #input number
    move $s1, $v0
    sw $s1, 0($t0)                          # $s1 = A[i]
    addi $t2, $t2, 1                        # i++
    mul $t1, $t2, 4                          # adress of A + 4 (A[i++])
    j input_array

CheckArray:
    li $t0, 0                               # i = 0
    addi $t1, $a1, 0                        # $t1 = address of A

loop:
    beq $t0, $s0, PrintResult                # if i = N in ket qua
    lw $v1, 0($t1)                          # lay gia tri o dia chi $t1 luu vao $v1(A[i])
    bltz $v1, FindMax                       # if $v1 < 0 FindMax

Continue:
    addi $t0, $t0, 1                        # i++
    addi $t1, $t1, 4                        # address of A + 4
    j loop                                  # return loop1

FindMax:
    addi $t5, $v1, 0                        # Max tam thoi
    addi $t2, $t1, 4                        # adress of A[i] + 4
    addi $t3, $t0, 1                        # j = i + 1

Max:

```

```

Max:
    beq $t3, $s0, Update      # if j = N tiếp tục vòng loop1
    lw $s1, 0($t2)            # lưu giá trị từ địa chỉ $t2 vào $s1
    bgez $s1, Skip            # if $s1 >= 0 Skip
    blt $s1, $t5, Skip        # if $s1 < $v1 Skip
    addi $t5, $s1, 0          # Max = Max mới
    addi $t4, $t2, 0          # vị trí Maxx

Skip:
    addi $t3, $t3, 1          # j++
    addi $t2, $t2, 4          # Address of A[j] + 4
    j Max                     # return loop 2

Update:
    sw $t5, 0($t1)            # truyền giá trị max vào A[i]
    sw $v1, 0($t4)            # truyền giá trị ở A[i] vào vị trí Max
    j Continue

PrintResult:
    li $v0, 4
    la $a0, Message2
    syscall
    li $t0, 0                 # i = 0

Print:
    beq $t0, $s0, exit        # if i = N exit
    li $v0, 1
    lw $a0, 0($a1)
    syscall                   # in ra A[i]
    li $v0, 4
    la $a0, Space
    syscall                   # in dấu cách
    addi $a1, $a1, 4
    addi $t0, $t0, 1          # i++
    j Print

PrintError:
    li $v0, 4
    la $a0, Error
    syscall
    j main

```

- Result

```

Nhap so phan tu mang: 8
Nhap so: -1
Nhap so: 4
Nhap so: -3
Nhap so: -4
Nhap so: -2
Nhap so: 2
Nhap so: -5
Nhap so: -3
Day so am giam ma khong lam thay doi thu tu so duong la:
-1 4 -2 -3 -3 2 -4 -5

```

- Phân tích cách thực hiện:

+ Sau khi nhập vào các phần tử của mảng ta tìm giá trị âm trong dãy sau đó tăng biến chạy j lên 1 so với biến chạy i đang đứng ở giá trị âm sau đó ta cho j chạy từ i+1 đến cuối dãy nếu như tìm được giá trị âm lớn nhất thì sẽ đổi chỗ giá trị lớn nhất cho A[i] sau đó tăng i lên rồi tiếp tục thực hiện cho đến khi i = kích thước của mảng

Assigment 4.C

- Code

```

.data
    String: .space 256
    Message: .asciiiz "Nhap vao xau ky tu: "
    Message1: .asciiiz "Tu co do dai dai nhat trong xau la: "

.text
main:
    li $v0, 4
    la $a0, Message
    syscall                                # Nhap vao xau
    li $v0, 8
    la $a0, String
    li $a1, 256
    syscall                                # input String
    la $a1, String                        # load address of String
    j LongestString                      # jump LongestString

exit:
    li $v0, 10
    syscall                                #eixt

LongestString:
    li $s0, 32    #space
    li $s1, 10    #\n
    li $s2, 0     #Start character
    li $s3, 0     #End character
    li $v0, 0     #count

Reset:
    addi $t2, $a1, 0                    # luu vi tri bat dau tam thoi vao $t2

Check:
    lb $v1, 0($a1)                      # lay chu cai tu $a1 ve luu vao $v1
    beqz $v1, PrintResult                # if c = '\0' In ket qua
    beq $v1, $s0, RisePointer            # if c = '\n' RisePointer
    beq $v1, $s1, RisePointer            # if c = ' ' RisePointer
    addi $a1, $a1, 1                    # dia chi A[i] + 1
    j Check

RisePointer:
    sub $t3, $a1, $t2                    # $t3 = vi tri cuoi - vi tri dau
    ble $t3, $v0, Skip                  # if $t3 < count skip
    addi $v0, $t3, 0                    # count = $t3
    addi $s2, $t2, 0                    # Start character = gia tri dia chi luu o $t2
    addi $s3, $a1, 0                    # End character = gia tri dia tri ket thuc

Skip:
    addi $a1, $a1, 1                    # A[i] + 1
    j Reset                            # Reset lai vi tri bat dau

PrintResult:
    li $v0, 4
    la $a0, Message1
    syscall                                # in ra ket qua

Print:
    beq $s2, $s3, exit                  # if dia chi bat dau = dia chi ket thuc eixt
    li $v0, 11
    lb $a0, 0($s2)
    lb $a0, 0($s2)
    syscall                                #in ra tu dai nhat
    addi $s2, $s2, 1
    j Print

```

- Result

```
Nhap vao xau ky tu: Hello world i am hero
Tu co do dai dai nhat trong xau la: Hello
-- program is finished running --
```

- *Phân tích cách thực hiện:*

+ Sau khi nhập vào chuỗi ta lưu giá trị địa chỉ đầu của chuỗi vào biến sau đó tăng địa chỉ và thu về các ký tự của chuỗi khi gặp ký tự ' ' thì so sánh vị trí cuối với biến lưu vị trí đầu nếu giá trị từ đi lớn hơn từ dài nhất hiện tại thì ta cập nhật lại từ dài nhất và vị trí đầu cuối, cho đến khi ta gặp ký tự kết thúc chuỗi thì dừng lại và in ra kết quả.