

BÁO CÁO THỰC HÀNH

| | | |
|---------------|-------------------|---------------|
| Họ tên | Trương Thanh Minh | Lớp: KHTN2021 |
| MSSV | 21520064 | STT: 6 |
| Bài Thực Hành | LAB04 | |
| CBHD | Trương Văn Cường | |

1 Yêu cầu thực hành

Điểm buổi thực hành

| | | |
|--------------------------|--|--|
| Chuyên cần (20%) | | |
| Trình bày (20%) | | |
| Nội dung thực hành (60%) | | |
| Câu 1: | | |
| Câu 2: | | |
| Câu 3: | | |
| Câu 4: | | |
| Câu 5: | | |
| Câu 6: | | |
| Tổng (100%) | | |

Bài tập thực hành:

Viết chương trình MIPS thực hiện các công việc sau:

1. Nhập số phần tử của mảng.
2. Nhập giá trị của mảng (Kiểm tra số dương).
3. In phần tử mảng.
4. Tính tổng các phần tử trong mảng.
5. Đếm số chẵn lẻ trong mảng.
6. Tìm min max trong mảng.

2 Thực hành

- Trước khi thực hiện các yêu cầu của đề bài, ta khởi tạo những biến cần thiết

```

        .data
string1: .asciiiz "Nhap so phan tu mang (be hon 100): "

pti: .asciiiz "Nhap phan tu thu"
xuatmang: .asciiiz "Mang ban vua nhap la:\n"
KhoangCach: .asciiiz " "
HaiCham: .asciiiz ": "

        .align 4
arr: .space 400 # Tao mang so nguyen gom 100 phan tu

string4: .asciiiz "Phan tu phai la so duong. Vui long nhap lai!\n"
string5: .asciiiz "So khong hop le, vui long nhap lai!\n"
string6: .asciiiz "Tong cua mang la: "
chan: .asciiiz "So so chan cua mang: "
le: .asciiiz "So so le cua mang: "
max: .asciiiz "So lon nhat trong mang: "
min: .asciiiz "So be nhat trong mang: "

XuongDong: .asciiiz "\n"
        .text

```

Figure 0: Khởi tạo ban đầu

1. Nhập số phần tử mảng

- Để nhập phần tử, ta sẽ nhập như bình thường. Sau đó, để kiểm tra điều kiện, ta tiến hành so sánh với cận trên và cận dưới của điều kiện. (Nếu không thỏa điều kiện nó sẽ yêu cầu nhập lại)

```

main:

addi $t5, $zero, 100 # t5 là giới hạn số phần tử của mảng
addi $t6, $zero, 1 # cần duy nhất

Nhapsort:
    # in thông báo nhập
    li $v0, 4
    la $a0, string1
    syscall

    li $v0, 5
    syscall
    move $t0, $v0 # t0 là số phần tử

    # kiểm tra điều kiện
    slt $t3, $t5, $t0 # 100 < t0
    slt $t4, $t0, $t6 # t0 < 1

    or $t3, $t3, $t4
    beq $t3, $zero, thoatnhapsort

    # nếu không thỏa điều kiện thì nhập lại số phần tử
    li $v0, 4
    la $a0, string5
    syscall

    j Nhapsort

thoatnhapsort:

```

Figure 1.1: Đoạn code nhập vào số phần tử của mảng thỏa điều kiện yêu cầu

- Demo kết quả thực thi chương trình

```

Nhập số phần tử mảng (bề hơn 100): 101
Số không hợp lệ, vui lòng nhập lại!
Nhập số phần tử mảng (bề hơn 100): -2
Số không hợp lệ, vui lòng nhập lại!
Nhập số phần tử mảng (bề hơn 100): 0
Số không hợp lệ, vui lòng nhập lại!
Nhập số phần tử mảng (bề hơn 100): 5
Nhập phần tử thứ 0:

```

Figure 1.2: Hình ảnh kết quả thực thi đoạn code trên

2. Nhập giá trị của mảng (Kiểm tra số dương)

- Đầu tiên, ta sẽ khởi tạo các thanh ghi cần thiết. Trong đó, $t1$ là biến đếm để kiểm tra xem đã nhập đủ số lượng các phần tử hay chưa, $t2$ là giá trị nhập vào của phần tử thứ i , $s5, s6$ lần lượt lưu lại số số chẵn của chương trình,

```

la $s0, arr # s0 là base của mảng

add $t1, $zero, $zero # t1 là biến i

addi $s7, $zero, 1 # để phục vụ cho việc xét chẵn lẻ
add $s5, $zero, $zero # số số chẵn
add $s6, $zero, $zero # số số lẻ

LoopNhap:
    beq $t1, $t0, ExitNhap # Nếu t1 = số phần tử mảng thì dừng

    Nhapt:
        # Nhập phần tử thứ i
        li $v0, 4
        la $a0, pti
        syscall

        li $v0, 4
        la $a0, KhoangCach
        syscall

        li $v0, 1
        la $a0, ($t1)
        syscall

        li $v0, 4
        la $a0, HaiCham
        syscall

        # Nhập phần tử thứ i
        li $v0, 5
        syscall

```

```

        move $t2, $v0 # t2 = A[i]

        # Kiểm tra số được nhập có phải là số không âm
        slt $t3, $zero, $v0 # 0 < v0 -> 1
        add $t6, $zero, $zero # gán t6 = 0
        addi $t6, $t6, 1 # t6 = 1
        beq $t3, $t6, ThoatNhap # > 0, sau đó nhập phần tử tiếp theo

        li $v0, 4
        la $a0, string4
        syscall

        j Nhappt

ThoatNhap:

        sw $t2, ($s0)

        addi $t1, $t1, 1 # tăng biến đếm i lên 1
        addi $s0, $s0, 4 # tăng giá trị base lên 4
        j LoopNhap

ExitNhap:
        li $v0, 4
        la $a0, KhoangCach
        syscall

```

Figure 2.1: Đoạn code thực hiện công việc nhập giá trị mảng thỏa điều kiện đề bài

- Sau đây là demo kết quả thực hiện chương trình

```

Nhap so phan tu mang (be hon 100): 5
Nhap phan tu thu 0: 7
Nhap phan tu thu 1: 0
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 1: 1
Nhap phan tu thu 2: 9
Nhap phan tu thu 3: -5
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 3: 2
Nhap phan tu thu 4: 4

```

Figure 2.2: Demo kết quả thực thi đoạn code trên

3. In phần tử mảng

- Đầu tiên, ta sẽ tiến hành khởi tạo các thanh ghi cần thiết.
- Sau đó, ta sẽ làm các bước tương tự như khi nhập phần tử.

```

la $s0, arr # s0 là base của mảng
add $t1, $zero, $zero # t1 là biến i

add $s3, $zero, $zero # max
add $s4, $zero, $zero # min

lw $t2, ($s0)
move $s3, $t2
move $s4, $t2

la $s0, arr # s0 là base của mảng
add $t1, $zero, $zero # t1 là biến i

add $t7, $zero, $zero # t7 là tổng của mảng
# In ra các phần tử trong mảng
li $v0, 4
la $a0, xuấtmang
syscall

```

Figure 3.1: Khởi tạo giá trị các thanh ghi cần thiết cho yêu cầu của bài toán

```

LoopXuat:
    beq $t1, $t0, ExitXuat # check nhập đủ phần tử hay chưa

    lw $t2, ($s0) # t2 = A[i]

    li $v0, 1
    la $a0, ($t2)
    syscall

    add $t7, $t7, $t2

    andi $t8, $t2, 1 # t2 = 0 thì nó là số chẵn
    beq $t8, $zero, SoChan # tính số số chẵn

    addi $s6, $s6, 1 # tính số số lẻ

    # tìm max min

    slt $t9, $t2, $s3 # if t2 < max -> cập nhật, 0 thì cập nhật
    bne $t9, $zero, ContinueXetMin
    move $s3, $t2

    slt $t9, $s4, $t2 # min < t2, 0 thì cập nhật
    bne $t9, $zero, Continue
    move $s4, $t2

    li $v0, 4
    la $a0, KhoangCach
    syscall

    addi $t1, $t1, 1 # tăng i lên 1
    addi $s0, $s0, 4 # tăng base lên 4
    j LoopXuat

ExitXuat:

```

Figure 3.2: Thực hiện xuất mảng

- Kết quả thực thi của đoạn chương trình trên

```

Nhap so phan tu mang (be hon 100): 5
Nhap phan tu thu 0: 7
Nhap phan tu thu 1: 0
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 1: 1
Nhap phan tu thu 2: 9
Nhap phan tu thu 3: -5
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 3: 2
Nhap phan tu thu 4: 4
Mang ban vua nhap la:
7 1 9 2 4

```

Figure 3.3: Demo kết quả của đoạn code trên

4. Tính tổng các phần tử trong mảng.

- Đầu tiên, ta sẽ thực hiện việc gán giá trị vào một thanh ghi ($t7$) để lưu lại giá trị của tổng các phần tử trong mảng.

```
add $t7, $zero, $zero # t7 là tổng của mảng
```

Figure 4.1: Khởi tạo thanh ghi

- Sau mỗi lần duyệt các phần tử trong mảng, ta sẽ thực hiện cộng dồn vào thanh ghi *t7*.

```
lw $t2, ($s0) # t2 = A[i]

li $v0, 1
la $a0, ($t2)
syscall

add $t7, $t7, $t2
```

Figure 4.2: Thực hiện tính tổng các phần tử trong mảng

- Sau khi thực hiện xong, ta tiến hành in tổng đó ra.

```
# in câu xuất tổng
li $v0, 4
la $a0, string6
syscall

# in tổng của mảng
li $v0, 1
move $a0, $t7
syscall

li $v0, 4
la $a0, XuongDong
syscall
```

Figure 4.3: Xuất giá trị tổng các phần tử mảng

- Để dễ hình dung hơn, ta sẽ nhìn vào kết quả sau khi thực hiện đoạn chương trình trên (*dòng cuối cùng là kết quả ta cần, các dòng trên chỉ để giúp ta dễ dàng hình dung hơn về input*).

```
Nhap so phan tu mang (be hon 100): 5
Nhap phan tu thu 0: 7
Nhap phan tu thu 1: 0
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 1: 1
Nhap phan tu thu 2: 9
Nhap phan tu thu 3: -5
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 3: 2
Nhap phan tu thu 4: 4
Mang ban vua nhap la:
7 1 9 2 4
Tong cua mang la: 23
```

Figure 4.4: Demo kết quả của đoạn code trên

5. Đếm số chẵn lẻ trong mảng.

- Cũng như ở trên, trước khi thực hiện việc đếm số chẵn lẻ ở trong mảng, ta sẽ tiến hành khởi tạo giá trị cho các thanh ghi: s5 sẽ lưu số số chẵn, s6 sẽ lưu số số lẻ và bên cạnh đó, ta thêm thanh ghi s7 để sau này xét tính chẵn lẻ cho các phần tử.

```
addi $s7, $zero, 1 # để phục vụ cho việc xét chẵn lẻ
add $s5, $zero, $zero # số số chẵn
add $s6, $zero, $zero # số số lẻ
```

Figure 5.1: Khởi tạo giá trị thanh ghi

- Tiếp theo, ta sẽ đếm số chẵn lẻ trong mảng

```
andi $t8, $t2, 1 # t2 = 0 thì nó là số chẵn
beq $t8, $zero, SoChan # tính số số chẵn

addi $s6, $s6, 1 # tính số số lẻ
```

Figure 5.2: Đoạn code thể hiện việc đếm số chẵn lẻ

- Ở đây, ta không cần để ý đến chỗ đoạn code (có comment là tìm max min).

```
SoChan:
    addi $s5, $s5, 1
    # tìm max min

    slt $t9, $t2, $s3 # if t2 < max -> cập nhật, 0 thì cập nhật
    bne $t9, $zero, ContinueXetMin
    move $s3, $t2

    slt $t9, $s4, $t2 # min < t2, 0 thì cập nhật
    bne $t9, $zero, Continue
    move $s4, $t2

    li $v0, 4
    la $a0, KhoangCach
    syscall

    addi $t1, $t1, 1 # tăng i lên 1
    addi $s0, $s0, 4 # tăng base lên 4
    j LoopXuat
```

Figure 5.3: Đoạn code mô tả các bước khi move tới SoChan

- Đoạn code xuất ra kết quả

```

# in so so chan cua mang
li $v0, 4
la $a0, chan
syscall

li $v0, 1
move $a0, $s5
syscall

li $v0, 4
la $a0, XuongDong
syscall

#in so so le
li $v0, 4
la $a0, le
syscall

li $v0, 1
move $a0, $s6
syscall

li $v0, 4
la $a0, XuongDong
syscall

```

Figure 5.4: Đoạn code xuất ra kết quả

- Cuối cùng, ta sẽ quan sát phần kết quả thực thi của đoạn code

```

Nhap so phan tu mang (be hon 100): 5
Nhap phan tu thu 0: 7
Nhap phan tu thu 1: 0
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 1: 1
Nhap phan tu thu 2: 9
Nhap phan tu thu 3: -5
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 3: 2
Nhap phan tu thu 4: 4
Mang ban vua nhap la:
7 1 9 2 4
Tong cua mang la: 23
So so chan cua mang: 2
So so le cua mang: 3

```

Figure 5.5: Kết quả demo

6. Tìm max min trong mảng.

- Bước đầu, ta vẫn sẽ khởi tạo giá trị của thanh ghi cần dùng (\$s3 lưu giá trị *max*, \$s4 lưu giá trị *min*).

```

add $s3, $zero, $zero # max
add $s4, $zero, $zero # min

```

Figure 6.1: Khởi tạo giá trị thanh ghi

- Sau đây sẽ là chi tiết cài đặt tìm max, min

```

# tim max min

slt $t9, $t2, $s3 # if t2 < max -> capnhat, 0 thi cap nhat
bne $t9, $zero, ContinueXetMin
move $s3, $t2

slt $t9, $s4, $t2 # min < t2, 0 thi cap nhat
bne $t9, $zero, Continue
move $s4, $t2

ContinueXetMin:
    slt $t9, $s4, $t2 # min < t2, 0 thi cap nhat
    bne $t9, $zero, Continue
    move $s4, $t2

    li $v0, 4
    la $a0, KhoangCach
    syscall

    addi $t1, $t1, 1 # tang i len 1
    addi $s0, $s0, 4 # tang base len 4
    j LoopXuat

Continue:
    li $v0, 4
    la $a0, KhoangCach
    syscall

    addi $t1, $t1, 1 # tang i len 1
    addi $s0, $s0, 4 # tang base len 4
    j LoopXuat

```

Figure 6.2: Chi tiết đoạn code tìm min max các phần tử trong mảng

- Đoạn code xuất ra kết quả, khá tương tự với các lần trên

```

# in so lon nhat trong mang
li $v0, 4
la $a0, max
syscall

li $v0, 1
move $a0, $s3
syscall

li $v0, 4
la $a0, XuongDong
syscall

# in so be nhat trong mang
li $v0, 4
la $a0, min
syscall

li $v0, 1
move $a0, $s4
syscall

```

Figure 6.3: Đoạn code xuất ra kết quả

- Sau đây là kết quả thực hiện chương trình

```
Nhap so phan tu mang (be hon 100): 5
Nhap phan tu thu 0: 7
Nhap phan tu thu 1: 0
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 1: 1
Nhap phan tu thu 2: 9
Nhap phan tu thu 3: -5
Phan tu phai la so duong. Vui long nhap lai!
Nhap phan tu thu 3: 2
Nhap phan tu thu 4: 4
Mang ban vua nhap la:
7 1 9 2 4
Tong cua mang la: 23
So so chan cua mang: 2
So so le cua mang: 3
So lon nhat trong mang: 9
So be nhat trong mang: 1
```

Figure 6.3: Demo kết quả thực thi