

BÁO CÁO THỰC HÀNH

Họ tên	Trần Hoàng Bảo Ly	Lớp: KHTN2021
MSSV	21521109	STT: 20
Bài Thực Hành	Lab 5	
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:		
Tổng (100%)		

Bài tập thực hành:

- Mô phỏng chương trình (*) MIPS Assembly trên MARS theo các yêu cầu sau.
 - Nhập mảng A gồm n ($0 < n \leq 100$) phần tử là các số nguyên dương.
 - Thực hiện sắp xếp mảng A theo 3 phương pháp.
 - Lựa chọn phương pháp sắp xếp để thực hiện sắp xếp mảng A
 - In mảng A sau khi sắp xếp ra màn hình

2 Thực hành

2.1 Mã MIPS Assembly

2.1.1 Khai báo và khởi tạo biến

```
#Tran Hoang Bao Ly
#21521109
#20
.data
    Array: .space 400
    nhapn: .asciiz "Nhap n la so phan tu cua mang nguyen duong (n>0 va
n<=100): "
    nhaploi: .asciiz "So n ban vua nhap khong phu hop (n<=0 hoac n>100), moi
nhap lai: "
    nhapso: .asciiz "Nhap phan tu thu "
    haicham: .asciiz ": "
    space: .asciiz " "
    nhapsoam: .asciiz "So ban vua nhap <=0, moi ban nhap lai: "
    inmang: .asciiz "After sorted \n A[] = "
    xuonghang: .asciiz "\n"
    bubbleprint: .asciiz "Thuc thien bubble sort \n"
    insertprint: .asciiz "Thuc hien insertion sort \n"
    selectprint: .asciiz "Thuc hien selecttion sort \n"
    luachon: .asciiz "Moi ban lua chon cach sort: \n 1:Bubble sort. \n 2:
Insertion sort. \n 3: Selection sort. \n (cac so khac mac dinh la selection
sort) \n"

.text
```

2.1.2 Nhập giá trị n, thỏa yêu cầu bài toán đặt ra

```
main:
    li $v0, 4 # write string
    la $a0, nhapn
    syscall

    li $v0, 5 #read integer
    syscall
    addi $t0, $v0, 0 #t0 = v0 (n = input())

    #Neu so vua nhap khong thoa dieu kien nhap lai

    WHILE:
        slti $t1, $t0, 1 #t1 = t0<1? 1 : 0
        addi $t2, $0, 100 #t2 = 100
        slt $t2, $t2, $t0 #t2 = 100<t0? 1: 0
        or $t1,$t1,$t2 #t1 = t1|t2

        beq $t1,$0, ENDWhile

        li $v0, 4 # write string
        la $a0, nhaploi
        syscall
        li $v0, 5 #read integer
        syscall
        addi $t0, $v0, 0 #t0 = v0

        j WHILE
    ENDWhile:
```

2.1.3 Nhập giá trị cho từng phần tử trong mảng, thỏa yêu cầu bài toán đặt ra

```
#Khoi tao mang va i = 0
la $s0, Array
addi $t1, $zero, 0
#Nhap gia tri cho tung phan tu
FOR:
    slt $t2,$t1, $t0 # t2 = t1<t0?1:0
    beq $t2,$0 ENDFOR #if (t2 == 0) break
    li $v0, 4 # write string
    la $a0, nhapso
    syscall
    li $v0, 1 #write int
    addi $a0, $t1, 1
    syscall

    li $v0, 4 # write string
    la $a0, haicham
    syscall

    li $v0, 5 #read int
    syscall
    addi $s1, $v0,0 #s0 = input() (from v0)
    #Neu so am, nhap lai so vua nhap
    AW:
        slti $t2, $s1, 1 # t2 = s1<0?1:0
        beq $t2, $0 ENDAW #if (t2 == 0) break;
        li $v0, 4 # write string
        la $a0, nhapsoam
        syscall
        li $v0, 5 #read int
        syscall
        addi $s1, $v0,0 #s0 = input() (from v0)
        J AW
    ENDAW:
    #Luu so vua nhap vao mang
    sll $s2, $t1,2 #s2 = t1*4
    add $s2, $s2, $s0
    sw $s1, ($s2)    #Arr[i] = s1;

    addi $t1,$t1,1
    j FOR
ENDFOR:
```

2.1.4 Cho phép người dùng lựa chọn cách Sắp xếp (bubble sort, insertion sort, selection sort)

```
li $v0, 4 # write string
la $a0, luachon
syscall
li $v0, 5 #read integer
syscall
addi $t1, $v0, 0 #t0 = v0 (n = input())
addi $t2, $0, 1
beq $t1,$t2, BUBBLESORT
addi $t2,$0,2
beq $t1,$t2, INSERTSORT

j SELECTSORT
```

2.1.5 Bubble sort

BUBBLESORT:

```
li $v0, 4 # write string
la $a0, bubbleprint
syscall

addi $t1, $0, 0
subi $s1, $t0, 1
#for (i,0,n-1)
FOR:
    slt $t3,$t1,$s1 # t3 = t1<s1?1:0
    beq $t3, $0, ENDFOR
    addi $t2, $zero, 0
    sub $s2, $t0, $t1
    subi $s2, $s2, 1 # size = n-i-1
    # print Arr[i]
    #for(j,0,n-i-1)
    FOR2:
        slt $t3, $t2, $s2
        beq $t3, $0, ENDFOR2
        sll $t3, $t2, 2
        add $t3, $t3, $s0 # t3 = pointer of A[j]
        lw $s3, ($t3) # s3 = A[j]
        lw $s4, 4($t3) # s4 = A[j+1]
        slt $t4, $s3, $s4 # t4 = s3 < s4? 1:0
        bne $t4, $0, ENDSWAP #if (A[i] > A[i+1]) swap
        sw $s4, ($t3)
        sw $s3, 4($t3)
        ENDSWAP:
        addi $t2, $t2, 1
        j FOR2
    ENDFOR2:
    addi $t1, $t1, 1
    j FOR
ENDFOR:
j PRINTARRAY
```

2.1.6 Insertion sort

```
#InSertSort
INSERTSORT:
li $v0, 4 # write string
la $a0, insertprint
syscall

addi $t1, $0, 0 # i = 0
subi $s1, $t0, 1 # n-1
#for(i,0,n-1)
IFOR:
    slt $t3,$t1,$s1 # t3 = t1<s1?1:0
    beq $t3, $0, IENDFOR
    addi $t2, $t1, 1 # j = i+1
    addi $s2, $t0, 0 #s2 = n
    #for(i,i+1,n)
    IFOR2:
        slt $t3, $t2, $s2
        beq $t3, $0, IENDFOR2
        sll $t3, $t1, 2
        add $t3, $t3, $s0 # t3 = pointer of A[i]
        lw $s3, ($t3) #s3 = A[i]
        sll $t4, $t2, 2
        add $t4, $t4, $s0 # t4 = pointer of A[j]
        lw $s4, ($t4) #s4 = A[j]
        slt $t5, $s4, $s3 # t5 = s4<s3?1:0
        beq $t5, $0, IENDSWAP #if (s4>s3) swap(s4,s3)
        sw $s3, ($t4)
        sw $s4, ($t3)
    IENDSWAP:
        addi $t2, $t2, 1
        j IFOR2
    IENDFOR2:
        addi $t1, $t1, 1
        j IFOR
IENDFOR:
j PRINTARRAY
```

2.1.7 Selection sort

```
#SelectionSort
SELECTSORT:
li $v0, 4 # write string
la $a0, selectprint
syscall
addi $t1, $0, 0 # i = 0
subi $s1, $t0, 1 # n-1
SFOR:
    slt $t3,$t1,$s1 # t3 = t1<s1?1:0
    beq $t3, $0, SENDFOR
    addi $t2, $t1, 1 # j = i+1
    addi $s2, $t0, 0 #s2 = n

    sll $t3, $t1, 2
    add $t3, $t3, $s0 # t3 = pointer of A[i]
    lw $s3, ($t3) #s3 = A[i]
    # Let s4 = s3 is a minimum value of A[i:n]
    addi $s4, $s3, 0
    addi $t4, $t3,0
    #for(i,i+1,n)
    SFOR2:
        slt $t5, $t2, $s2
        beq $t5, $0, SENDFOR2
        #find minimum value of A[i:n]
        sll $t5, $t2, 2
        add $t5, $t5, $s0 # t5 = pointer of A[j]
        lw $s5, ($t5) #s5 = A[j]
        slt $t6, $s5, $s4 # t6 = s5<s4?1:0
        beq $t6, $0, SENDSWAP #if (s4>s5) update s4
        addi $s4, $s5,0
        addi $t4, $t5,0
        SENDSWAP:
        addi $t2, $t2,1
        j SFOR2
    SENDFOR2:
        #swap A[i] and minimum value of A[i:n]
        sw $s3, ($t4)
        sw $s4, ($t3)
        addi $t1, $t1, 1
        j SFOR
    SENDFOR:
    j PRINTARRAY
```


2.1.8 In mảng sau khi sắp xếp

```
#print array
PRINTARRAY:
li $v0, 4
la $a0, inmang
syscall
addi $t1, $0, 0
FORP:
    slt $t2, $t1, $t0 # t2 = t1<t0?1:0
    beq $t2, $0, ENDFORP
    # print Arr[i]
    lw $t2, ($s0)
    li $v0, 1
    addi $a0, $t2, 0
    syscall
#print space
li $v0, 4
la $a0, space
syscall
addi $s0, $s0, 4
addi $t1, $t1, 1
j FORP
ENDFOR:
```

2.2 Kết quả thực hiện chương trình

2.2.1 Thực hiện nhập n

```
Nhap n la so phan tu cua mang nguyen duong (n>0 va n<=100): 101
So n ban vua nhap khong phu hop (n<=0 hoac n>100), moi nhap lai: -2
So n ban vua nhap khong phu hop (n<=0 hoac n>100), moi nhap lai: 0
So n ban vua nhap khong phu hop (n<=0 hoac n>100), moi nhap lai: 8
```

Hình 2-1. Kết quả thực hiện chương trình (nhập n)

Chương trình chạy đúng với mong muốn, thỏa yêu cầu bài toán đặt ra ($0 < n \leq 100$).

2.2.2 Thực hiện nhập giá trị cho từng phần tử trong mảng

```
Nhap phan tu thu 1: -1
So ban vua nhap <=0, moi ban nhap lai: 50
Nhap phan tu thu 2: 2
Nhap phan tu thu 3: 16
Nhap phan tu thu 4: 9
Nhap phan tu thu 5: 7
Nhap phan tu thu 6: 24
Nhap phan tu thu 7: 3
Nhap phan tu thu 8: 10
```

Hình 2-2 Kết quả thực hiện chương trình (Nhập từng phần tử trong mảng)

Chương trình chạy đúng với mong muốn, thỏa yêu cầu bài toán đặt ra ($A[i] > 0 \forall i \in n$).

2.2.3 Thực hiện lựa chọn thuật toán sắp xếp.

```
Moi ban lua chon cach sort:
1:Bubble sort.
2: Insertion sort.
3: Selection sort.
(cac so khac mac dinh la selection sort)
1
```

Hình 2-3 chương trình in ra thông báo lựa chọn

Chương trình in ra thông báo lựa chọn (nhập 1, lựa chọn thuật toán bubble sort)

2.2.4 Chương trình in mảng kết quả ra màn hình.

```
Thuc thien bubble sort
After sorted
A[] = 2 3 7 9 10 16 24 50
```

Hình 2-4 chương trình in mảng đã sắp xếp ra màn hình

2.2.5 Thực hiện tương tự với các mảng khác nhau và lần lượt chọn 2, 3 và -1

```
2
Thuc hien insertion sort
After sorted
A[] = 1 4 5 7 8 9 11 20
```

Hình 2-5 in mảng đã sắp xếp ra màn hình khi lựa chọn insertion sort

```
Thuc hien selection sort
After sorted
A[] = 1 2 2 6 7 14 18 50
```

Hình 2-6 in mảng đã sắp xếp ra màn hình khi lựa chọn selection sort

```
-1
Thuc hien selection sort
After sorted
A[] = 1 2 3 5 7 19 42 50
```

Hình 2-7 in mảng đã sắp xếp khi nhập -1 (selection sort mặc định)

Kết luận: Chương trình trên chạy đúng với dự đoán, thỏa yêu cầu chương trình (*) đặt ra.

3 Bài nộp

Bài nộp gồm 2 file, 1 file báo cáo .pdf và 1 file .zip gồm tất cả các file liên quan đến bài thực hành (LAB5)

Toàn bộ file nguồn được lưu trữ trên github, github link: [Click here](#)