**BÁO CÁO THỰC HÀNH**

|  |  |  |
| --- | --- | --- |
| Họ tên | Lê Thanh Minh | Lớp: KHTN2021 |
| MSSV | 21520063 | STT: 5 |
| Bài Thực Hành | LAB4 | |
| CBHD | **Trương Văn Cương** | |

# Yêu cầu thực hành

**Về phần trình bày:**

* Sinh viên trình bày đúng theo định dạng báo cáo mà CBHD đưa ra.
* Cần chú thích bảng, hình (nếu có).
* Sử dụng chức năng Insert Caption và Cross-reference cho chú thích Bảng, Hình
* Sử dụng tính năng Screenshot để chụp kết quả mô phỏng.

**Quy trình thực hành:**

* Sinh viên chuẩn bị bài ở nhà, và có mặt đúng giờ tại phòng LAB.
* Sinh viên thực hành theo hướng dẫn, và nộp bài đúng hạn.
* Hoàn thành bài tập về nhà (nếu có)
* Tất cả các bài báo cáo có hành vi sao chép của nhau sẽ bị **điểm 0**

**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:**

**Sử dụng kiến thức về Assembly đã học để thực hiện yêu cầu sau:**

1. Nhập mảng gồm n phần tử (0<n<101), mỗi phần tử có giá trị dương
2. Sắp xếp lại các phần tử trong mảng theo thứ tự (tăng dần hoặc giảm dần), ít nhất 1 thuật toán sort và nhiều nhất 3 thuật toán sort

# Thực hành

## Cơ sở lý thuyết các thuật toán sắp xếp

* **Selection sort**: Luôn thay thế các phần tử đã chọn bằng phần tử nhỏ nhất
* **Insertion sort**: Liên tục chèn các phần tử nhỏ hơn vào mảng đã sắp xếp
* **Bubble sort**: Liên tục so sánh các phần tử kề nhau và sắp xếp

Link lý thuyết :[Sort theory](https://www.geeksforgeeks.org/comparison-among-bubble-sort-selection-sort-and-insertion-sort/)

Minh họa cách hoạt động của 3 loại sort trên : [Sorting visualization](https://visualgo.net/en/sorting)

## Chuẩn bị dữ liệu

# Le Thanh Minh KHTN2021

# 21520063

.data #base data

array:      .space 440

string1:    .asciiz "Moi ban nhap so phan tu cua mang (<=100): "

string3:    .asciiz "So phan tu khong hop le ,vui long nhap lai (<=100): "

string4:    .asciiz "Nhap phan tu thu "

string5:    .asciiz ": "

string6:    .asciiz " "

string7:    .asciiz "Cac phan tu cua mang lan luot la: "

string8:    .asciiz "Phan tu vua nhap <0, moi nhap lai: "

string9:    .asciiz "\nSo cac phan tu le va chan lan luot la: "

string10:   .asciiz "\nGia tri lon nhat trong mang la : "

string11:   .asciiz "\nGia tri nho nhat trong mang la : "

string12:   .asciiz "\nTong gia tri cac phan tu trong mang la: "

## Viết lệnh cấu trúc

### Nhập mảng

.text #instruction

    la $s0,array #array

    li $v0,4

    la $a0,string1

    syscall #print

    li $v0,5

    syscall #load input

    #check if element <0 or >100

    addi $t2,$v0,0 #number of elements

    slti $t3,$t2,1

    bne $t3,$0,InputElementNumber

    subi $t4,$t2,101

    slti $t3,$t4,0

    beq $t3,$0,InputElementNumber

    #if condition is true, start to get the input of element

    bne $t3,$0,Elements

InputElementNumber:

    li $v0,4

    la $a0,string3

    syscall #print

    li $v0,5

    syscall #load input

    addi $t2,$v0,0

    slti $t3,$t2,1

    bne $t3,$0,InputElementNumber

    subi $t4,$t2,101 #t4=t2-50

    slti $t3,$t4,0  #if t4>0 then t3=0 , else t3=1 to check whether t4 exceed 50 or not

    beq $t3,$0,InputElementNumber

Elements:

    addi $t3,$0,0 #

While: #nhap phan tu thu i

    slt $t4,$t3,$t2

    beq $t4,$0, Check

    li $v0,4

    la $a0,string4

    syscall #print

    li $v0, 1

    addi $a0, $t3,1 #input number ith element

    syscall

    li $v0, 4

    la $a0,string5

    syscall #print

    li $v0,5

    syscall #load input

    add $t5,$0,$v0 #input=t5

ForinWhile:

    slti $t6,$t5,1

    beq $t6,$0,Out

    li $v0, 4

    la $a0,string8

    syscall #print

    li $v0, 5

    syscall

    addi $t5,$v0,0

    j ForinWhile

Out:

    sll $s1, $t3, 2

    add $s1,$s0,$s1

    sw $t5,0($s1)

    addi $t3,$t3,1

    j While

InputElementNumber:

    li $v0,4

    la $a0,string3

    syscall #print

    li $v0,5

    syscall #load input

    addi $t2,$v0,0

    slti $t3,$t2,1

    bne $t3,$0,InputElementNumber

    subi $t4,$t2,101 #t4=t2-50

    slti $t3,$t4,0  #if t4>0 then t3=0 , else t3=1 to check whether t4 exceed 50 or not

    beq $t3,$0,InputElementNumber

Elements:

    addi $t3,$0,0 #

While: #nhap phan tu thu i

    slt $t4,$t3,$t2

    beq $t4,$0, Check

    li $v0,4

    la $a0,string4

    syscall #print

    li $v0, 1

    addi $a0, $t3,1 #input number ith element

    syscall

    li $v0, 4

    la $a0,string5

    syscall #print

    li $v0,5

    syscall #load input

    add $t5,$0,$v0 #input=t5

ForinWhile:

    slti $t6,$t5,1

    beq $t6,$0,Out

    li $v0, 4

    la $a0,string8

    syscall #print

    li $v0, 5

    syscall

    addi $t5,$v0,0

    j ForinWhile

Out:

    sll $s1, $t3, 2

    add $s1,$s0,$s1

    sw $t5,0($s1)

    addi $t3,$t3,1

    j While

\*Chú thích :

|  |  |
| --- | --- |
| InputElementNumber | Nếu số phần tử của mảng nhập vào nhỏ hơn 1 hoặc lớn hơn 100 thì phải nhập lại |
| Elements | Khi Thỏa số phần tử thì bắt đầu nhập các phần tử trong mảng |
| While | Nhập phần tử đến khi đủ số phần tử và kiểm tra các phần tử trong mảng có bé hơn hoặc bằng 0 hay không, kết thúc bằng Check |
| ForinWhile | Ở trong While, nếu phần tử nhập vào bé hơn hoặc bằng 0 thì nhập lại, nếu không tiếp tục phần While |
| Out | Khi phần tử thỏa điều kiện lớn hơn 0 thì tiếp tục quay lại lệnh While |

### In từng phần tử trong mảng

Check:

    li $v0, 4

    la $a0,string7

    syscall #print

    addi $t3,$0,0 #counting element

    addi $s2,$s0,0 # temp store address of s0

ForinPrint1:

    slt $t4,$t3,$t2

    beq $t4,$0, Sort

    lw $s1,0($s2) #get element

    li $v0, 1

    addi $a0, $s1,0#print number ith element

    syscall

    li $v0, 4

    la $a0,string6

    syscall #print

    addi $t3,$t3,1  #changeelement

    addi $s2,$s2,4

    j ForinPrint1

\*Chú thích :

|  |  |
| --- | --- |
| Check | Khởi tạo lại các phần tử để in mảng |
| ForinPrint1 | In tất cả phần tử trong mảng trước khi được sắp xếp, nếu đã in hết thì bắt đầu sắp xếp bằng cách nhảy xuống khối lệnh Sort |

## Các thuật toán sort sử dụng assembly

### Bubble Sort

### 

Sort:

#initialize

    li $v0, 4

    la $a0,string9

    addi $s2,$s0,0

    subi $t2,$t2,1

    syscall #print

    addi $t3,$0,-1 #counting element first loop

Forsort1:

    addi $t3,$t3,1

    addi $t5,$0,0 #t5 was current first comparasion

    slt $t4,$t3,$t2

    beq $t4,$0,After

Forsort2:

    slt $t4,$t5,$t2

    beq $t4,$0,Forsort1

    sll $t5,$t5,2

    # load two mediate parameter

    add $s2,$s0,$t5

    lw $t8,0($s2)

    lw $t7,4($s2)

    slt $t4,$t7,$t8

    bne $0,$t4,Swap

Continue2:

    srl $t5,$t5,2

    addi $t5,$t5,1

    j Forsort2

Swap:

    addi $s4,$t7,0

    sw ,$t8,4($s2)

    sw, $s4,0($s2)

    j Continue2

After:

    addi $s2,$s0,0

    addi $t3,$0,0

    addi $t2,$t2,1

ForinPrint2:

    slt $t4,$t3,$t2

    beq $t4,$0, Exit

    lw $s1,0($s2) #get element

    li $v0, 1

    addi $a0, $s1,0#print number ith element

    syscall

    li $v0, 4

    la $a0,string6

    syscall #print

    addi $t3,$t3,1  #changeelement

    addi $s2,$s2,4

    j ForinPrint2

Exit:

### Insertion Sort

Sort:

#initialize

    li $v0, 4

    la $a0,string9

    addi $s2,$s0,0

    syscall #print

    addi $t3,$0,0 #counting element first loop

Forsort1:

    addi $t3,$t3,1

    addi $t5,$t3,1 #t5 was current first comparasion

    slt $t4,$t3,$t2

    beq $t4,$0,After

Forsort2:

    subi $t5,$t5,1

    sll $t5,$t5,2

    # load two mediate parameter

    add $s2,$s0,$t5

    lw $t8,0($s2)

    lw $t7,-4($s2)

    slt $t4,$t8,$t7

    bne $0,$t4,Swap

Continue2:

    srl $t5,$t5,2

    bne $0,$t4,Forsort2

    j Forsort1

Swap:

    addi $s4,$t7,0

    sw ,$t8,-4($s2)

    sw, $s4,0($s2)

    j Continue2

After:

    addi $s2,$s0,0

    addi $t3,$0,0

    addi $t2,$t2,1

ForinPrint2:

    slt $t4,$t3,$t2

    beq $t4,$0, Exit

    lw $s1,0($s2) #get element

    li $v0, 1

    addi $a0, $s1,0#print number ith element

    syscall

    li $v0, 4

    la $a0,string6

    syscall #print

    addi $t3,$t3,1  #changeelement

    addi $s2,$s2,4

    j ForinPrint2

Exit:

### Selection sort

Sort:

#initialize

    li $v0, 4

    la $a0,string9

    addi $s2,$s0,0

    syscall #print

    addi $t3,$0,-1 #counting element first loop

Forsort1:

    addi $t3,$t3,1

    addi $t5,$t3,0 #t5 was current first comparasion

    addi $t6,$t3,0

    slt $t4,$t3,$t2

    beq $t4,$0,After

Forsort2:

    slt $t4,$t5,$t2

    beq $t4,$0,Forsort1

    sll $t5,$t5,2

    sll $t6,$t6,2

    # load two mediate parameter

    add $s2,$s0,$t5

    add $s3,$s0,$t6

    lw $t8,0($s2)

    lw $t7,0($s3)

    slt $t4,$t8,$t7

    bne $0,$t4,Swap

Continue2:

    srl $t5,$t5,2

    srl $t6,$t6,2

    addi $t5,$t5,1

    j Forsort2

Swap:

    addi $s4,$t7,0

    sw ,$t8,0($s3)

    sw, $s4,0($s2)

    j Continue2

\*Chú thích

|  |  |  |  |
| --- | --- | --- | --- |
| Khối Lệnh | Bubble Sort | Selection Sort | Insertion Sort |
| Sort | Khởi tạo các phần tử cần thiết cho các thuật toán sort | | |
| Forsort1 | Cập nhật giá trị của Forsort2, nếu chạy đủ n-1 vòng sẽ tiến đến khối lệnh After | Cập nhật lại giá trị của Forsort1 và cập nhật lại cả giá trị của Forsort2 (giảm số vòng lặp) của forsort2, nếu hoàn thành vòng lặp sẽ tiến tới khối lệnh After | Cập nhật lại giá trị và tiến hành Forsort2,Nếu không sẽ tiến tới khối lệnh After |
| Forsort2 | Cập nhật lại vị trí phần tử. Nếu phần tử không sắp xếp đúng thứ tự sẽ nhảy tới khối lệnh Swap, nếu đúng sẽ tiếp tục vòng lặp Forsort2, nếu đủ số vòng sẽ nhảy đến Forsort1 | | Chạy và nếu phần tử không sắp xếp đúng thứ tự sẽ nhảy tới khối lệnh Swap, nếu không sẽ kết thúc vòng lặp và nhảy đến Forsort1 |
| Continue2 | Tiếp tục vòng lệnh Forsort2 | | |
| Swap | Thay đổi giá trị của hai biến trong mảng, sau đó nhảy về khối lệnh Continue2 | | |

### In ra kết quả

After:

    addi $s2,$s0,0

    addi $t3,$0,0

ForinPrint2:

    slt $t4,$t3,$t2

    beq $t4,$0, Exit

    lw $s1,0($s2) #get element

    li $v0, 1

    addi $a0, $s1,0#print number ith element

    syscall

    li $v0, 4

    la $a0,string6

    syscall #print

    addi $t3,$t3,1  #changeelement

    addi $s2,$s2,4

    j ForinPrint2

Exit:

## Kết quả thực nghiệm

Khi nhập số phần tử không hợp lệ thì phải nhập lại đến khi hợp lệ :

Text

Description automatically generated

Nếu nhập phần tử thứ i không hợp lệ sẽ phải nhập lại :

Text, letter

Description automatically generated

Trong trường hợp nhập đủ sẽ in ra kết quả như sau :

Text

Description automatically generated

Ta thấy chương trình thực hiện đúng với yêu cầu được đưa ra

* Bài làm gồm 2 file :

1. Là file PDF ghi lại kết quả báo cáo có tên là 21520063-lab05.pdf
2. Một file zip lưu lại tất cả các file liên quan

* Toàn bộ file được tải lên github ở link sau : [Lab05-IT012](https://github.com/Kryst4lize/Lab05-IT012)