

BÁO CÁO THỰC HÀNH

Họ tên	Lương Toàn Bách	Lớp: KHTN2021
MSSV	21521845	STT: 23
Bài Thực Hành	LAB-05	
CBHD	Trương Văn Cường	

1. ĐỀ BÀI:

- Cho mảng a có n phần tử, với $1 \leq n \leq 100$. Phần tử thứ i mang giá trị nguyên dương a_i .
- Yêu cầu: Với ngôn ngữ lập trình Assembly, hãy nhập mảng, kiểm tra tính hợp lệ về dữ liệu của mảng và sắp xếp lại mảng.

2. TỔNG QUAN

- Bài thực hành trên ngôn ngữ lập trình Assembly với ba tính năng:
 - Sắp xếp tăng dần theo thuật toán Interchange Sort.
 - Sắp xếp tăng dần theo thuật toán Bubble Sort.
 - Sắp xếp giảm dần theo thuật toán Gnome Sort.

3. CƠ SỞ LÝ THUYẾT

4. THỰC HÀNH

4.1 Nhập kích thước mảng

```
input:
    li $v0, 4
    la $a0, inputMessage
    syscall

    li $v0, 5
    syscall

    move $t0, $v0
    bgt $t0, 100, reInput
    ble $t0, 0, reInput

    li $t1, 0
    li $t2, 0
    j inputLoop
reInput:
    li $v0, 4
    la $a0, reInputMessage
    syscall
    j input
inputLoop:
    beq $t1, $t0, choseType
    li $v0, 5
    syscall
```

#Xuat ra inputMessage

#Nhap so phan tu cua mang

#Neu so nhap vao >100 || <0 thi reInput

#t1 dung de luu bien dem
#t2 dung de luu bien index

#Nhap lai

#Dieu kien dung

4.2 Nhập giá trị các phần tử của mảng và

4.3 Chọn phương pháp sắp xếp mảng

```
j input
inputLoop:
    beq $t1, $t0, choseType           #Điều kiện dừng
    li $v0, 5
    syscall

    ble $v0, 0, inputElement
    sw $v0, myArray($t2)              #Luu du lieu vao mang

    addi $t1, $t1, 1
    addi $t2, $t2, 4
    j inputLoop
choseType:
    li $v0, 4
    la $a0, chooseTypeMessage
    syscall

    li $v0, 5
    syscall
    move $t9, $v0
    beq $t9, 1, interchangeSort
    beq $t9, 2, bubbleSort
    beq $t9, 3, gnomeSort
    j exit
```

4.4 Thuật toán InterchangeSort

4.4.1 Mã giả

```
void interchangeSort(int a[], int size)
{
    for (int i = 0; i < arr.length - 1; i++) {
        for (int j = i + 1; j < arr.length; j++) {
            if (arr[i] > arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}
```

4.4.2 Cài đặt bằng Assembly

```

interchangeSort:
    li $t1, 0
    li $t2, 0
    subi $t5, $t0, 1
    j interchangeLoop1
interchangeLoop1:
    beq $t1, $t5, output
    move $t3, $t1
    move $t4, $t2
    addi $t3, $t3, 1
    addi $t4, $t4, 4
    j interchangeLoop2
interchangeLoop2:
    beq $t3, $t0, endInterchangeLoop2
    lw $t6, myArray($t2)
    lw $t7, myArray($t4)
    bge $t7, $t6, interchangeElse
    interchangeSwap:
        sw $t6, myArray($t4)
        sw $t7, myArray($t2)
        j interchangeElse
    interchangeElse:
        addi $t3, $t3, 1
        addi $t4, $t4, 4
        j interchangeLoop2
    endInterchangeLoop2:
        addi $t1, $t1, 1
        addi $t2, $t2, 4
    j interchangeLoop1

```

4.5 Thuật toán BubbleSort

4.5.1 Mã giả

```
// hàm sắp xếp nổi bọt (bubble sort)
void BubbleSort(int a[], int n){
    int temp; // biến tạm temp
    for (int i = 0; i < n; i++){
        for (int j = i + 1; j < n; j++){
            if (a[j] > a[j+1]){
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
}
```

4.5.2 Cài đặt bằng Assembly

```
bubbleSort:
    subi $t1, $t0, 1
    sll $t2, $t1, 2
    j bubbleLoop1
bubbleLoop1:
    bltz $t1, output
    li $t3, 0
    li $t4, 0
    j bubbleLoop2
bubbleLoop2:
    beq $t3, $t1, endBubbleLoop2
    addi $t5, $t4, 4
    lw $t6, myArray($t4)
    lw $t7, myArray($t5)
    ble $t6, $t7, bubbleElse
    j swap
swap:
    sw $t6, myArray($t5)
    sw $t7, myArray($t4)
    j bubbleElse
bubbleElse:
    addi $t3, $t3, 1
    addi $t4, $t4, 4
    j bubbleLoop2
endBubbleLoop2:
    addi $t1, $t1, -1
    addi $t2, $t2, -4
    j bubbleLoop1
```

4.6 Thuật toán GnomeSort

4.6.1 Mã giả

```
void gnomeSort(int arr[], int n)
{
    int index = 0;

    while (index < n) {
        if (index == 0)
            index++;
        if (arr[index] >= arr[index - 1])
            index++;
        else {
            swap(arr[index], arr[index - 1]);
            index--;
        }
    }
    return;
}
```

4.6.2 Cài đặt bằng Assembly

```
gnomeSort:
    li $t1, 0
    li $t2, 0
    j loop1
loop1:
    beq $t1, $t0, output
    bne $t1, $zero, endIf1
    addi $t1, $t1, 1
    addi $t2, $t2, 4
    j endIf1
endIf1:
    subi $t3, $t2, 4
    lw $t4, myArray($t2)
    lw $t5, myArray($t3)
    blt $t4, $t5, else2
    addi $t1, $t1, 1
    addi $t2, $t2, 4
    j loop1
else2:
    sw $t4, myArray($t3)
    sw $t5, myArray($t2)
    addi $t1, $t1, 1
    addi $t2, $t2, 4
    j loop1
```

4.6 Xuất kết quả

```
output:
    li $v0, 4
    la $a0, outputMessage
    syscall
    li $t1, 0
    li $t2, 0
    j outputLoop
outputLoop:
    beq $t1, $t0, exit
    li $v0, 1
    lw $t7, myArray($t2)
    move $a0, $t7
    syscall

    li $v0, 4
    la $a0, printSpace
    syscall

    addi $t1, $t1, 1
    addi $t2, $t2, 4
    j outputLoop
exit:
```

#Tang bien dem len 1
#Tang bien index len 4

5. THỰC NGHIỆM


```
Reset: reset completed.

Nhap so phan tu cua mang: 4
1
7
2
5
1.Interchangesort
2.BubbleSort
3.InterchangeSort
Nhap loai sort:2
Mang da duoc sap xep la:
1 2 5 7
-- program is finished running (dropped off bottom) --

Nhap so phan tu cua mang: 5
1
5
3
9
3
1.Interchangesort
2.BubbleSort
3.InterchangeSort
Nhap loai sort:1
Mang da duoc sap xep la:
1 3 3 5 9
-- program is finished running (dropped off bottom) --
```

```
Reset: reset completed.  
  
Nhap so phan tu cua mang: 4  
1  
9  
2  
6  
1.Interchangesort  
2.BubbleSort  
3.InterchangeSort  
Nhap loai sort:3  
Mang da duoc sap xep la:  
1 2 6 9  
-- program is finished running (dropped off bottom) --
```