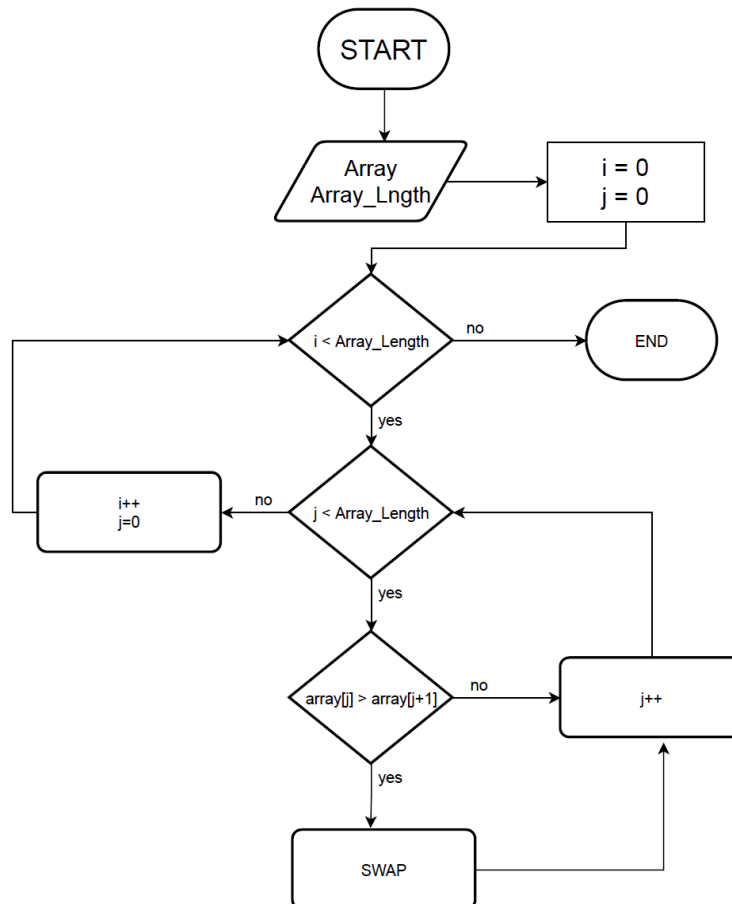


Software 4  
Arrays in Assembly  
Summer 2021  
Thiha Myint

### Description

In this software lab, we tried to gain more experience for writing assembly language programs. The lab is to design an assembly program that implements a bubble sort on a 16-element array of data.

### Flowchart



## Verification

```
1 .data
2 Array: .word 1, 121, 43, 11, 2, 4, 7, 99, 100, 101, 6, 11, 5, 12, 23, 66
```

Fig 1: Input Assigned

Registers Memory				
Address	+0	+1	+2	+3
0x10000030	5	0	0	0
0x1000002c	11	0	0	0
0x10000028	6	0	0	0
0x10000024	101	0	0	0
0x10000020	100	0	0	0
0x1000001c	99	0	0	0
0x10000018	7	0	0	0
0x10000014	4	0	0	0
0x10000010	2	0	0	0
0x1000000c	11	0	0	0
0x10000008	43	0	0	0
0x10000004	121	0	0	0
0x10000000	1	0	0	0

Jump to

Display Settings

Fig 2: Input in memories

Registers

Memory

Address	+0	+1	+2	+3
0x10000030	99	0	0	0
0x1000002c	66	0	0	0
0x10000028	43	0	0	0
0x10000024	23	0	0	0
0x10000020	12	0	0	0
0x1000001c	11	0	0	0
0x10000018	11	0	0	0
0x10000014	7	0	0	0
0x10000010	6	0	0	0
0x1000000c	5	0	0	0
0x10000008	4	0	0	0
0x10000004	2	0	0	0
0x10000000	1	0	0	0

Jump to

-- choose --

Up

Down

Display Settings

Decimal

Fig 3: Sorted after run

## Assembly code

```
.data
Array: .word 1, 121, 43, 11, 2, 4, 7, 99, 100, 101, 6, 11, 5, 12, 23, 66

.text
main:
    la    a0, Array
    li    a1, 15

    jal   ra, bubble_sort
    j     end

bubble_sort:
    addi   sp, sp, -32
    sw     ra, 28(sp)
    sw     s0, 24(sp)
    addi   s0, sp, 32
    sw     a0, -12(s0)
    sw     a1, -16(s0)
    mv     a0, zero
    sw     a0, -24(s0)
    j      For_i

For_i:
    lw     a0, -24(s0)
    lw     a1, -16(s0)
    bge    a0, a1, sort_done
    mv     a0, zero
    sw     a0, -28(s0)
    j      For_j

For_j:
    lw     a0, -28(s0)
    lw     a1, -16(s0)
    bge    a0, a1, inc_i

if_greater:
    lw     a0, -12(s0)
    lw     a1, -28(s0)
    slli   a1, a1, 2
    add    a0, a0, a1
    lw     a1, 0(a0)
    lw     a0, 4(a0)
    bge    a0, a1, inc_j

swap:
    lw     a0, -12(s0)
    lw     a1, -28(s0)
    slli   a1, a1, 2
    add    a0, a0, a1
    lw     a0, 0(a0)
    sw     a0, -20(s0)
    lw     a0, -12(s0)
    lw     a1, -28(s0)
    slli   a1, a1, 2
```

```
    add    a0, a0, a1
    lw     a1, 4(a0)
    sw     a1, 0(a0)
    lw     a0, -20(s0)
    lw     a1, -12(s0)
    lw     a2, -28(s0)
    slli   a2, a2, 2
    add    a1, a1, a2
    sw     a0, 4(a1)
    j      inc_j
inc_j:
    lw     a0, -28(s0)
    addi   a0, a0, 1
    sw     a0, -28(s0)
    j      For_j
inc_i:
    lw     a0, -24(s0)
    addi   a0, a0, 1
    sw     a0, -24(s0)
    j      For_i
sort_done:
    lw     s0, 24(sp)
    lw     ra, 28(sp)
    addi   sp, sp, 32
    ret
end:
    addi   a0, x0, 10
    ecall
```