

# Bài thực hành số 6

Lớp : 139365 – Học phần: Thực hành Kiến trúc máy tính

Họ và tên : Nguyễn Thị Thùy Dung

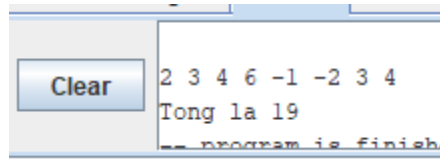
MSSV : 20215009

## Bài 1:

```
1  .data
2  A: .word 2,3,4,6,-1,-2,3,4      # khai tạo mảng
3  message: .asciiz " "
4  message1: .asciiz "\nTong la "
5  .text
6  main:
7      la $a0,A
8      li $a1,8
9      j mspfx
10     nop
11
12  mspfx:
13      addi $v0,$zero,0
14      addi $v1,$zero,0
15      addi $t0,$zero,0
16      addi $t1,$zero,0
17  loop:
18      add $t2,$t0,$t0
19      add $t2,$t2,$t2
20
21      add $t3,$t2,$a0
22      lw  $t4,0($t3)
23      add $t1,$t1,$t4
24      slt $t5,$v1,$t1
25      bne $t5,$zero,mdfy
26      j test
27  mdfy:
28      addi $v0,$t0,1
29      addi $v1,$t1,0
30  test:
31      addi $t0,$t0,1
32      slt $t5,$t0,$a1
33      bne $t5,$zero,loop
34  done:  j continue
35  mspfx_end:
36  continue:
37  end_of_main:
```

Thực hiện gõ chương trình với công cụ MARS

- Kết quả :



- Giải thích :
  - Khai báo (dòng 2-4): Khai báo mảng A có 8 phần tử
  - Dòng 7-8 : lấy giá trị phần tử đầu tiên của mảng A, khởi tạo  $n = 8$  (số phần tử của mảng bằng 8).
  - Khởi tạo (dòng 13-16) : khởi tạo độ dài chuỗi tiền tố lớn nhất bằng 0, tổng của chuỗi tiền tố bằng 0, chỉ số index  $i$  bằng 0, sum bắt đầu chạy bằng 0
  - Thực hiện vòng lặp: lấy địa chỉ của từng phần tử của mảng thông qua \$t3 và lấy giá trị của mảng vào \$t4 (lw), cộng giá trị của A[i] (\$t4) vào sum (\$t1). So sánh sum đang có (\$t1) với max sum (\$v1). Nếu max sum (\$v1) nhỏ hơn sum (\$t1) thì cập nhật độ dài chuỗi tiền tố (\$v0) +1, max sum mới (\$v1) được cập nhật. Nếu không thì thực hiện so sánh xem index  $i$  (\$t0) với  $n$  (\$a1). Nếu  $i > n$  thì thoát vòng lặp, nếu không thực hiện vòng lặp. Max sum và độ dài chuỗi tiền tố cuối cùng chính là giá trị cần tìm

## Bài 2:

```
1  .data
2  A:  .word -5, 4, 3, 8, 10, 23, -10, 2, 3, 5, -10
3  Aend: .word
4  message: .asciiz "  "
5  .text
6  main:
7      la $a0, A
8      la $a1, Aend
9      addi $a1, $a1, -4
10     j sort
11  end_main:
12
13  sort:
14      beq $a0, $a1, done
15      j max
16  after_max:
17      lw $t0, 0($a1)
18      sw $t0, 0($v0)
19      sw $v1, 0($a1)
20
21      addi $a1, $a1, -4
22      j sort
23  done:  j after_sort
24
25  max:
26      addi $v0, $a0, 0
27      lw $v1, 0($v0)
28      addi $t0, $a0, 0
29
30  loop:
31      beq $t0, $a1, ret
32      addi $t0, $t0, 4
33      lw $t1, 0($t0)
34      slt $t2, $t1, $v1
35      bne $t2, $zero, loop
36      addi $v0, $t0, 0
37      addi $v1, $t1, 0
38      j loop
39  ret:
```

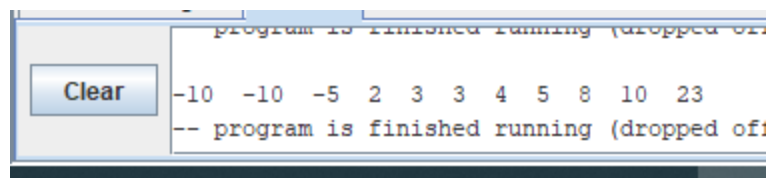
```

38 j after_max
39 after_sort:
40     la $s0,A
41     li $t0,0
42     li $s1,11
43 loops :
44     add $t2,$t0,$t0
45     add $t2,$t2,$t2
46     add $t3,$t2,$s0
47     lw $t4,0($t3)
48     li $v0,1
49     move $a0,$t4
50     syscall
51     li $v0,4
52     la $a0,message
53     syscall
54     addi $t0,$t0,1
55     slt $t5,$t0,$s1
56     bne $t5,$zero,loops

```

Thực hiện gõ chương trình với công cụ MARS

- Kết quả :



- Giải thích :

- Khai báo : Mảng A
- Ở đây lấy địa chỉ phần tử đầu tiên mảng A và địa chỉ cuối mảng A ta lấy thông qua biến Aend bởi vì bộ nhớ được cấp phát liên tiếp nên lấy địa chỉ Aend – 4 sẽ ra địa chỉ cuối của A
- Kiểm tra địa chỉ cuối của A có trùng địa chỉ đầu không. Nếu trùng thì mảng đã được sắp xếp, nếu chưa thì tiếp tục sắp xếp mảng. Thực hiện vòng lặp để tìm giá trị lớn nhất ở phần chưa được sắp xếp, sau đó đổi chỗ nó về cuối trước phần đã được sắp xếp. Sau mỗi vòng lặp thì địa chỉ cuối của mảng chưa được sắp xếp sẽ được giảm dần -4 . Cứ thực hiện như vậy n-1 lần ta được mảng đã được sắp xếp như kết quả.

### Bài 3:

```
1  # THUAT TOAN BUBBLE SORT CODE C
2
3  #   for (i = 0; i < n - 1; i++)
4  #       for (j = 0; j < n - i - 1; j++)
5  #           if (arr[j] > arr[j + 1])
6  #               swap(&arr[j], &arr[j + 1]);
7
8  .data
9  A: .word 0,0,10,-2,4,5,1,2,3,-10,11      # Khai bao mang A
10 message: .asciiz " "
11
12 .text
13     la $s0,A                             # dia chi cua A
14     li $t0,10                             # n- 1 = 9
15     li $t1,0                             # i =0
16 for1:
17     slt $t7,$t1,$t0                       # kiem tra i < n-1 ?
18     beq $t7,$zero,end_for1               # neu lon hon thoat vong for
19     li $t2,0                             # j =0
20
21     sub $t3,$t0,$t1                       # n-i-1
22 for2:
23     slt $t7,$t2,$t3                       # kiem tra j< n- i-1
24     beq $t7,$zero,end_for2               # neu lon hon thoa for2
25     sll $t4,$t2,2                         # t2 * 4
26     add $t5,$t4,$s0                       # t5 = t4 + s0 add+ add of A
27     lw $t6,0($t5)                        # lay a[j]
28     lw $s6,4($t5)                        # lay a[j+1]
29     slt $s5,$s6,$t6                       # so sanh a[i] va a[j+1]
30     bne $s5,$zero,swap                   #
31 j next
32 swap:
33     sw $t6,4($t5)
34     sw $s6,0($t5)
35 next:
36     addi $t2,$t2,1                       # j++
37 j for2
38 end for2:
```

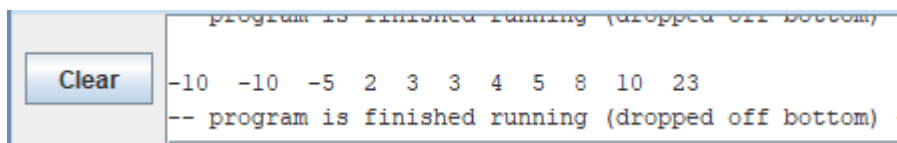
```

38         addi $t1,$t1,1           # i ++
39     j for1
40 end_for1:
41
42         la $s0,A                 # in mang A ra man hinh
43         li $s1,11
44         li $s2,0
45 loop:
46         slt $s3,$s2,$s1
47         beq $s3,$zero,end_loop
48         sll $s4,$s2,2
49         add $s5,$s0,$s4
50         lw $s6,0($s5)
51         li $v0,1
52         move $a0,$s6
53         syscall
54         li $v0,4
55         la $a0,message
56         syscall
57
58         addi $s2,$s2,1
59         j loop
60 end_loop:

```

### Thực hiện gõ chương trình với công cụ MARS

- Kết quả :



```

program is finished running (dropped off bottom)
-10 -10 -5 2 3 3 4 5 8 10 23
-- program is finished running (dropped off bottom) -

```

- Giải thích :

- Khai báo : Mảng A
- Thực hiện duyệt mảng từ 0 cho đến n-1 (for1). Với mỗi vòng lặp như vậy ta thực hiện vòng lặp for2 cho j chạy từ 0 đến n-1-i. Với mỗi lần lặp thực hiện so sánh a[j] có lớn hơn a[j+1] hay không. Nếu lớn hơn thì thực hiện đổi chỗ hai số đó cho nhau qua lệnh lw và sw.
- Sau đó thực hiện in mảng ra màn hình (dòng 42-59)
- Kết quả cho ra 1 mảng đã được sắp xếp theo kiểu Bubble sort

## Bài 4:

```
1  # THUAT TOAN INSERTION SORT CODE C
2
3  #void insertionSort(int arr[], int n)
4  #{
5  #     int i, key, j;
6  #     for (i = 1; i < n; i++)
7  #     {
8  #         key = arr[i];
9  #         j = i - 1;
10 #         while (j >= 0 && arr[j] > key)
11 #         {
12 #             arr[j + 1] = arr[j];
13 #             j = j - 1;
14 #         }
15 #         arr[j + 1] = key;
16 #     }
17 #}
18
19
20 .data
21 A: .word -1,10,2,3,5,1,2,3,5,5 # khai bao mang
22 message: .asciiz "          " # khai bao message
23 .text
24     la $s0,A
25     li $s1,10                    # n = 10
26     li $s2,1                    # i = 1
27 for:
28     slt $s3,$s2,$s1              # so sanh i<n
29     beq $s3,$zero,end_for       # neu lon hon end for
30     sll $s4,$s2,2                # i*4
31     add $s5,$s4,$s0              # s5 = 4i + add of a
32     lw $s6,0($s5)               # key = arr[i]
33     subi $s7,$s2,1              # j = i-1
34 loop:
35     sle $t0,$zero,$s7           # kiem tra 0 <= j
36     beq $t0,$zero,end_loop      # neu sai thi end loop
37     sll $t1,$s7,2                # j*4
```

```

37     sll $t1,$s7,2           # j*4
38     add $t2,$t1,$s0         # t2 = t1 + s0 load add
39     lw $t3,0($t2)           # lay arr[j]
40     slt $t4,$s6,$t3         # so sanh key voi arr[j]
41     beq $t4,$zero,end_loop  # neu key > arr[j] end loop
42     sw $t3,4($t2)           # arr[j+1] = a[j]
43     subi $s7,$s7,1          # j = j-1
44 j loop
45 end_loop:                   # endloop
46     sw $s6,4($t2)
47     addi $s2,$s2,1          # i++
48 j for
49 end_for:
50
51     la $s0,A                # In cac phan tu sau khi sap xep ra r
52     li $s1,10
53     li $s2,0
54 loops:
55     slt $s3,$s2,$s1

```

```

50
51     la $s0,A                # I
52     li $s1,10
53     li $s2,0
54 loops:
55     slt $s3,$s2,$s1
56     beq $s3,$zero,end_loops
57     sll $s4,$s2,2
58     add $s5,$s0,$s4
59     lw $s6,0($s5)
60     li $v0,1
61     move $a0,$s6
62     syscall
63     li $v0,4
64     la $a0,message
65     syscall
66     addi $s2,$s2,1
67     j loops

```

### Thực hành chương trình bằng công cụ MARS

- Kết quả :

```

program is finished running (dropped off)

Clear -10 -10 -5 2 3 3 4 5 8 10 23
-- program is finished running (dropped off)

```



- Giải thích :

Thực hiện vòng lặp for cho  $i$  chạy từ 1 đến  $n$ . Với mỗi vòng lặp sẽ thực hiện gán  $key = a[i]$  thông qua  $lw$ . Cho  $j = i-1$ , thực hiện vòng lặp loop, so sánh các điều kiện nếu  $j < 0$  hoặc  $a[j] \leq key$  thì thoát vòng lặp loop. Nếu không thì thực hiện gán  $a[j+1] = a[j]$  bằng lệnh `sw`. Thực hiện cập nhật  $j = j - 1$ . Sau đó thoát vòng lặp loop và cập nhật  $a[j+1] = key$ . Thực hiện vòng lặp for cho khi  $i = n$  thì dừng  
Kết quả cho ra mảng đã sắp xếp theo kiểu insertion sort.