

# BÁO CÁO THỰC HÀNH TUẦN 6

Nguyễn Thị Minh Châu

20214997

## Bài 1:

```
lab6_1aa
1  .data
2  A: .word -2, 6, -1, 3, -2
3  msg: .asciiz " "
4  msg1: .asciiz "\nTong cua chuoai tien to do la: "
5  msg2: .asciiz "Chuoai tien to co tong lon nhat la: "
6  .text
7  main: la $a0,A
8  li $a1,5
9  j mspfx
10 nop
11
12 mspfx:
13     addi $v0,$zero,0 #initialize length in $v0 to 0
14     addi $v1,$zero,0 #initialize max sum in $v1 to 0
15     addi $t0,$zero,0 #initialize index i in $t0 to 0
16     addi $t1,$zero,0 #initialize running sum in $t1 to 0
17 loop:
18     add $t2,$t0,$t0 #put 2i in $t2
19     add $t2,$t2,$t2 #put 4i in $t2
20     add $t3,$t2,$a0 #put 4i+A (address of A[i]) in $t3
21     lw $t4,0($t3) #load A[i] from mem(t3) into $t4
22     add $t1,$t1,$t4 #add A[i] to running sum in $t1
23     slt $t5,$v1,$t1 #set $t5 to 1 if max sum < new sum
24     bne $t5,$zero,mdfy #if max sum is less, modify results
25     j test #done?
26 mdfy:
27     addi $v0,$t0,1 #new max-sum prefix has length i+1
28     addi $v1,$t1,0 #new max sum is the running sum
```

- Dòng 2: Khai báo mảng A = {-2; 6; -1; 3; -2}
- Dòng 3, 4, 5: Khai báo 3 chuỗi
- Dòng 7: Gán địa chỉ mảng A vào a0
- Dòng 8: Gán giá trị a1 = 5 ( số phần tử của mảng)
- Dòng 12-16: Khởi tạo v0=0 là độ dài chuỗi tiền tố, v1=0 là tổng max, t0=0 (i=0), t1=0 là tổng

- Bắt đầu vòng lặp
  - Dòng 18-20: Gán địa chỉ của A[i] vào t3
  - Dòng 21: Gán giá trị của A[i] vào t4
  - Dòng 22: tính tổng t1=t1+t4 ( cộng thêm A[i] vào tổng)
  - Dòng 23: so sánh v1 < t1 ( so sánh tổng max < tổng)
  - Dòng 24: Nếu đúng thì chạy mdify, sai thì chạy test
- Mdfy
  - Dòng 27: Độ dài chuỗi cần tìm = i + 1
  - Dòng 28: tổng max mới = t1

```

29 test:
30     addi $t0,$t0,1 #advance the index i
31     slt $t5,$t0,$a1 #set $t5 to 1 if i<n
32     bne $t5,$zero,loop #repeat if i<n
33 done: j continue
34 mspfx_end:
35 continue:
36 end_of_main:
37     move $s1,$v0
38     la $s0,A
39     li $t0,0
40     li $v0, 4
41     la $a0, msg2
42     syscall
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,msg
53     syscall
54     addi $t0,$t0,1
55     slt $t5,$t0,$s1
56     bne $t5,$zero,loops
57     li $v0,4
58     la $a0,msg1
59     syscall
60     li $v0,1
61     move $a0,$v1
62     syscall

```

- test
  - Dòng 30:  $i=i+1$
  - Dòng 31: so sánh  $i < n$
  - Dòng 32: nếu đúng thì tiếp tục lặp, nếu sai thì jump đến continue
- In ra chuỗi và tổng
  - Dòng 37: Gán v0 vào s1 ( độ dài chuỗi cần tìm)
  - Dòng 38-39: gán địa chỉ mảng A vào s0 và gán t0=0 ( $i=0$ )
  - Dòng 40-41: in msg2
  - Dòng 43-56: thực hiện vòng lặp in ra chuỗi số cần tìm và xen kẽ dấu cách
  - Dòng 57-62: in ra msg1 và tổng của chuỗi cần tìm

Ta có kết quả như sau:

```
Chuoi tien to co tong lon nhat la: -2 6 -1 3
Tong cua chuoi tien to do la: 6
-- program is finished running (dropped off bottom) --
```

## Bài 2:

```
1  .data
2  A: .word 6,7,8,9,10,-3,-5,99,100,2,5,-98,34
3  Aend: .word
4  message: .asciiz " "
5  .text
6  main: la $a0,A #$a0 = Address(A[0])
7  la $a1,Aend
8  addi $a1,$a1,-4 #$a1 = Address(A[n-1])
9  j sort #sort
10 end_main:
11 sort: beq $a0,$a1,done #single element list is sorted
12 j max #call the max procedure
13 after_max: lw $t0,0($a1) #load last element into $t0
14 sw $t0,0($v0) #copy last element to max location
15 sw $v1,0($a1) #copy max value to last element
16 addi $a1,$a1,-4 #decrement pointer to last element
17 j sort #repeat sort for smaller list
18 done: j after_sort
19 max:
20 addi $v0,$a0,0 #init max pointer to first element
21 lw $v1,0($v0) #init max value to first value
22 addi $t0,$a0,0 #init next pointer to first
```

- Dòng 2-4: khai báo mảng A, Aend và chuỗi message
- Dòng 6-7: Gán địa chỉ của mảng A vào a0 và địa chỉ của Aend vào a1
- Dòng 8: Gán địa chỉ của A[n-1] vào a1 bằng cách lấy địa chỉ Aend-4
- Sort
  - Dòng 11: so sánh nếu a1=a0 (vị trí đầu và cuối của mảng) thì nhảy đến done, không thì nhảy đến max
- max
  - Con trỏ trỏ đến địa chỉ A[0]
  - Gán giá trị lớn nhất tạm thời là giá trị A[0]

```

23 loop:
24 beq $t0,$a1,ret #if next=last, return
25 addi $t0,$t0,4 #advance to next element
26 lw $t1,0($t0) #load next element into $t1
27 slt $t2,$t1,$v1 #(next)<(max) ?
28 bne $t2,$zero,loop #if (next)<(max), repeat
29 addi $v0,$t0,0 #next element is new max element
30 addi $v1,$t1,0 #next value is new max value
31 j loop #change completed; now repeat
32 ret:
33 j after_max

```

- Dòng 23-33: Bắt đầu lặp để tìm ra phần tử lớn nhất trong mảng, nhảy đến after\_max
- After\_max
  - Dòng 13-17: Đổi chỗ phần tử lớn nhất với phần tử cuối cùng trong mảng và cập nhật lại mảng mới. jump đến sort để kiểm tra vị trí đầu có bằng vị trí cuối chưa và tiếp tục thực hiện cho đến khi vị trí đầu bằng vị trí cuối sẽ jump đến done ( done jump đến after-sort)

```

33 j after_max
34 after_sort:
35 la $s0,A
36     li $t0,0
37     li $s1,13
38 loops :
39     add $t2,$t0,$t0
40     add $t2,$t2,$t2
41     add $t3,$t2,$s0
42     lw $t4,0($t3)
43     li $v0,1
44     move $a0,$t4
45     syscall
46     li $v0,4
47     la $a0,message
48     syscall
49     addi $t0,$t0,1
50     slt $t5,$t0,$s1
51     bne $t5,$zero,loops
52 li $v0, 10 #exit
53 syscall

```

- After\_sort
- Thực hiện vòng lặp in ra chuỗi đã sắp xếp

```

Mars Messages Run I/O
-- program is finished running --
-98 -5 -3 2 5 6 7 8 9 10 34 99 100
-- program is finished running --
Clear

```

### Bài 3:

```

1 .data
2 A: .word 6,11,59,-99,-32,10,44,3,4,2,3,4 # Khai bao mang A
3 message: .asciiz " "
4
5 .text
6 la $s0,A # dia chi cua A
7 li $t0,11 # n-1 = 11
8 li $t1,0 # i =0
9 for1:
10     slt $t7,$t1,$t0 # kiem tra i < n-1 ?
11     beq $t7,$zero,end_for1 # neu lon hon thoat vong for
12     li $t2,0 # j =0
13     sub $t3,$t0,$t1 # n-i-1
14 for2:
15     slt $t7,$t2,$t3 # kiem tra j < n-i-1
16     beq $t7,$zero,end_for2 # neu lon hon thoa for2
17     sll $t4,$t2,2 # t2 * 4
18     add $t5,$t4,$s0 # t5 = t4 + s0 add+ add of A
19     lw $t6,0($t5) # lay a[j]
20     lw $s6,4($t5) # lay a[j+1]
21     slt $s5,$s6,$t6 # so sanh a[i] va a[j+1]
22     bne $s5,$zero,swap #
23 j next
24 swap:
25     sw $t6,4($t5)
26     sw $s6,0($t5)
27

```

- Dòng 1-3: khai báo mảng A và chuỗi message “ “
- Dòng 5-8: Gán địa chỉ mảng A vào s0, gán giá trị t0=11, t1=0 (i=0)
- for1
  - Dòng 10: so sánh i < n-1
  - Dòng 11: Sai thì kết thúc vòng for
  - Dòng 12-13: Gán t2=0 (j=0), t3 = t0 – t1 ( = n-i-1)
- For2
  - Dòng 15: so sánh j < n-i-1
  - Dòng 16: Sai thì kết thúc vòng for
  - Dòng 17-22: so sánh A[j] và A[j+1] nếu A[j] > A[j+1] thì thực hiện swap ( đổi chỗ cho nhau)

```

27 next:
28     addi $t2,$t2,1           # j++
29 j for2
30 end_for2:
31     addi $t1,$t1,1           # i ++
32 j for1
33 end_for1:
34
35     la $s0,A                 # in mang A ra man hinh
36     li $s1,12
37     li $s2,0
38 loop:
39     slt $s3,$s2,$s1
40     beq $s3,$zero,end_loop
41     sll $s4,$s2,2
42     add $s5,$s0,$s4
43     lw $s6,0($s5)
44     li $v0,1
45     move $a0,$s6
46     syscall
47     li $v0,4
48     la $a0,message
49     syscall
50     addi $s2,$s2,1
51     j loop
52 end_loop:

```

- Tăng j tiếp tục chạy for2, tăng i tiếp tục chạy for1
- Dòng 35-52: thực hiện vòng lặp in mảng đã sắp xếp ra màn hình

```

-- program is finished running (dropped off bottom) --

-99  -32  2  3  3  4  4  6  10  11  44  59
-- program is finished running (dropped off bottom) --

```

## Bài 4:

```
1  .data
2  A: .word -6,20,2,4,5,68,2,3,45,5      # khai bao mang
3  message: .asciiz "      "           # khai bao message
4  .text
5      la $s0,A
6      li $s1,10                        # n = 10
7      li $s2,1                        # i = 1
8  for:
9      slt $s3,$s2,$s1                  # so sanh i<n
10     beq $s3,$zero,end_for            # neu lon hon end for
11     sll $s4,$s2,2                    # i*4
12     add $s5,$s4,$s0                  # s5 = 4i + add of a
13     lw $s6,0($s5)                   # key = arr[i]
14     subi $s7,$s2,1                  # j = i-1
15 loop:
16     sle $t0,$zero,$s7                # kiem tra 0 <= j
17     beq $t0,$zero,end_loop           # neu sai thi end loop
18     sll $t1,$s7,2                    # j*4
19     add $t2,$t1,$s0                  # t2 = t1 + s0 load add
20     lw $t3,0($t2)                   # lay arr[j]
21     slt $t4,$s6,$t3                  # so sanh key voi arr[j]
22     beq $t4,$zero,end_loop           # neu key > arr[j] end loop
23     sw $t3,4($t2)                   # arr[j+1] = a[j]
24     subi $s7,$s7,1                  # j = j-1
25 j loop
26 end_loop:                            # endloop
27
28     sw $s6,4($t2)
29     addi $s2,$s2,1                  # i++
30 j for
31 end_for:
32
33     la $s0,A                        # In cac phan tu sau khi sap xep ra
34     li $s1,10
35     li $s2,0
36 loops:
37     slt $s3,$s2,$s1
38     beq $s3,$zero,end_loops
39     sll $s4,$s2,2
40     add $s5,$s0,$s4
41     lw $s6,0($s5)
42     li $v0,1
43     move $a0,$s6
44     syscall
45     li $v0,4
46     la $a0,message
47     syscall
48     addi $s2,$s2,1
49     j loops
50 end_loops:
```

- Khai báo mảng và gán các giá trị cần dùng
- Thực hiện vòng for và bắt đầu lặp từ phần tử thứ hai của mảng cho đến khi gặp phần tử bé hơn, tại phần tử bé hơn quay ngược

lại bằng một biến đếm j, đẩy dần cho  $A[j-1]=A[j]$  cho đến khi nào mà  $A[j-1] \leq A[i]$  hoặc j chạy đến đầu của mảng ( $j=0$ ) thì ta cho  $A[i]$  nằm vào vị trí của  $A[j]$ .

- Dòng 31-49: in chuỗi đã sắp xếp ta có kết quả như sau:

```
-- program is finished running (dropped off bottom) --  
-6      2      2      3      4      5      5      20      45      68  
-- program is finished running (dropped off bottom) --
```