BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 6

Họ và tên: Nguyễn Quý Đức MSSV: 20235682

Mã HP: IT3280 Mã lớp: 156788

1. Assignment 1

w6h1.ASM*

```
.data
   A: .word -2, 6, -1, 3, -2
.text
main:
   la a0, A
   li a1, 5
   j mspfx
continue:
exit:
   li a7, 10
ecall
end_of_main:

mspfx:
   li s0, 0 # initialize length of prefix-sum in s0 to 0
   li s1, 0x80000000 # initialize max prefix-sum in s1 to smallest int
   li t0, 0 # initialize index for loop i in t0 to 0
   li t1, 0 # initialize running sum in t1 to 0
loop:
   add t2, t0, t0 # put 2i in t2
   add t2, t2, t2 # put 4i in t2
   add t3, t2, a0 # put 4i+A (address of A[i]) in t3
   lw t4, 0(t3) # load A[i] from mem(t3) into t4
   add t1, t1, t4 # add A[i] to running sum in t1
   blt s1, t1, mdfy # if (s1 < t1) modify results
   j next
mdfy:
   addi s0, t0, 1 # new max-sum prefix has length i+1
   addi s1, t1, 0 # new max sum is the running sum
next:
   addi t0, t0, 1 # advance the index i
   blt t0, a1, loop # if (i<n) repeat
done:
   j continue
mspfx_end:
```

Thanh ghi a0 = 0x10010000, chứa địa chỉ của array A; a1 = 5 là số phần tử của A; t2 = 4i là số byte off-set so với a0 để tính địa chỉ A[i]

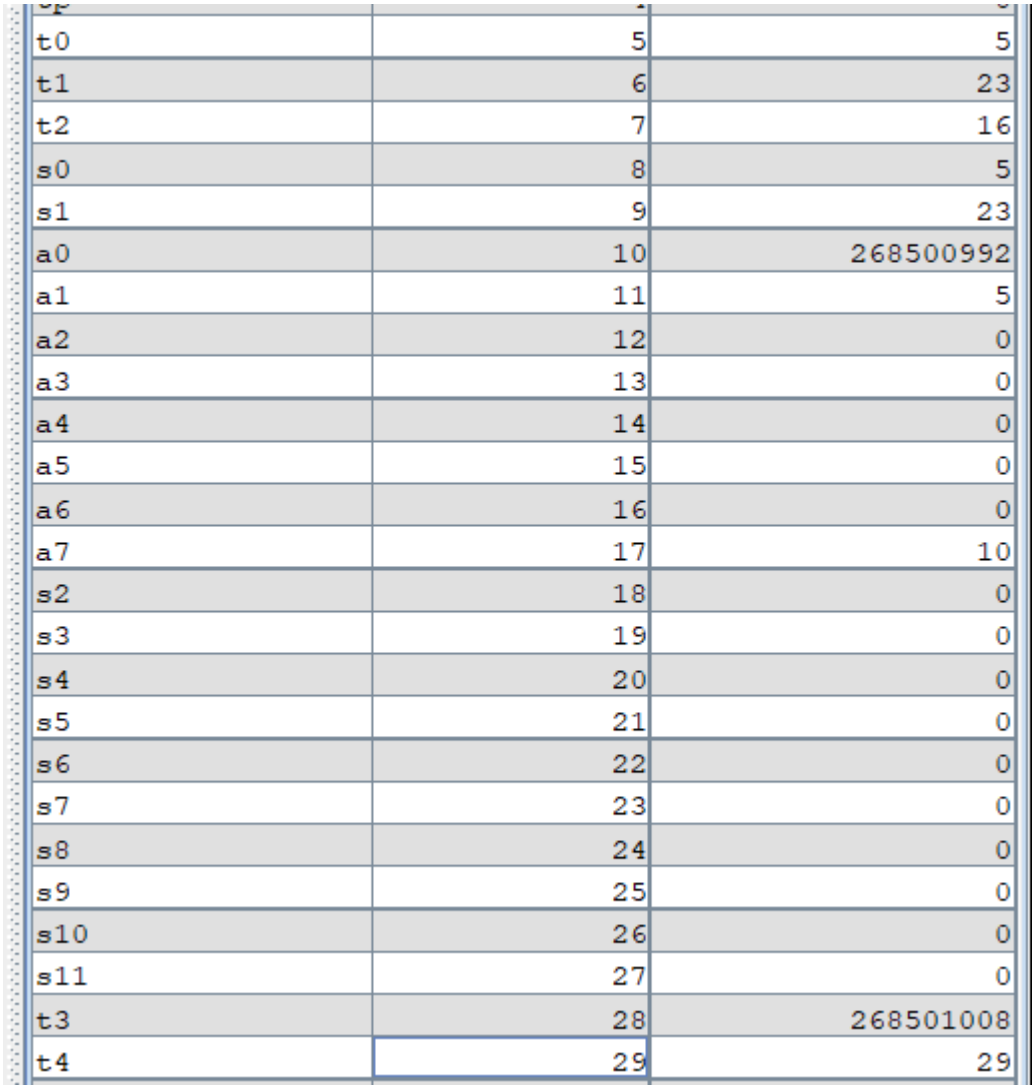
t0 (i)	t3 (A[i] addr)	t4 (A[i])	t1 (sum)	s1 (max sum)	s0 (length)
0	0x10010000	-2	-2	-2	1
1	0x10010004	6	4	4	2
2	0x10010008	-1	3	4	2
3	0x1001000c	3	6	6	4
4	0x10010010	-2	4	6	4

t0	5	5
t1	6	4
t2	7	16
s0	8	4
s1	9	6
a0	10	268500992
a1	11	5
a2	12	0
a3	13	0
a4	14	0
a5	15	0
a6	16	0
a7	17	10
s2	18	0
s3	19	0
s4	20	0
s5	21	0
s6	22	0
s7	23	0
s8	24	0
s9	25	0
s10	26	0
s11	27	0
t3	28	268501008
t4	29	-2

Với bộ số với A = [-2, -3, -1, 0, 29]

```
1  .data
2     A: .word -2, -3, -1, 0, 29
3  .text
4  main:
5     la a0, A
6     li a1, 5
7     j mspfx
```

t0 (i)	t3 (A[i] addr)	t4 (A[i])	t1 (sum)	s1 (max sum)	s0 (length)
0	0x10010000	-2	-2	-2	1
1	0x10010004	-3	-5	-2	1
2	0x10010008	-1	-6	-2	1
3	0x1001000c	0	-6	-2	1
4	0x10010010	29	23	23	5



t0	5	5
t1	6	23
t2	7	16
s0	8	5
s1	9	23
a0	10	268500992
a1	11	5
a2	12	0
a3	13	0
a4	14	0
a5	15	0
a6	16	0
a7	17	10
s2	18	0
s3	19	0
s4	20	0
s5	21	0
s6	22	0
s7	23	0
s8	24	0
s9	25	0
s10	26	0
s11	27	0
t3	28	268501008
t4	29	29

2. Assignment 2

```
.data
      A: .word 7, -2, 5, 1, 5, 6,
7, 3, 6, 8, 8, 59, 5
      Aend: .word
      # Aend duoc cap phat dia chi
ngay sau A
      nl: .asciz "\n"
      space: .asciz " "
      msg: .asciz "th loop: "
.text
main:
      la a2, A
      la a1, Aend
      addi a1, a1, -4
      li a5, 1   #loop counter
      j sort

after_sort:
      li a7, 10
      ecall

end_main:

sort:
      beq a2, a1, done
      j max

after_max:
      lw t0, 0(a1)
      sw t0, 0(s0)
      sw s1, 0(a1)
      addi a1, a1, -4

      j print_array

done:
      j after_sort

max:
      addi s0, a2, 0
      lw s1, 0(s0)
      addi t0, a2, 0

loop:
      beq t0, a1, ret
      addi t0, t0, 4
      lw t1, 0(t0)
      blt t1, s1, loop
      addi s0, t0, 0
      addi s1, t1, 0
      j loop

ret:
      j after_max

print_array:
      mv a0, a5
      li a7, 1
      ecall

      la a0, msg    # "nth loop: "
      li a7, 4
      ecall

      la a3, A
      la a4, Aend
      addi a4, a4, -4

print_loop:
      lw a0, 0(a3)
      li a7, 1
      ecall

      la a0, space
      li a7, 4
      ecall

      addi a3, a3, 4
      ble a3, a4, print_loop

print_last:
      la a0, nl
      li a7, 4
      ecall

      addi a5, a5, 1   # counter++
      j sort
```

A trước khi được sort:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+1
0x10010000	7	-2	5	1	
0x10010020	6	8	8	59	

A sau khi được sort:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	-2	1	3	5	5	5	6	6
0x10010020	7	7	8	8	59	0	0	0

Run I/O

```
1th loop: 7 -2 5 1 5 6 7 3 6 8 8 5 59
2th loop: 7 -2 5 1 5 6 7 3 6 8 5 8 59
3th loop: 7 -2 5 1 5 6 7 3 6 5 8 8 59
4th loop: 7 -2 5 1 5 6 5 3 6 7 8 8 59
5th loop: 6 -2 5 1 5 6 5 3 7 7 8 8 59
6th loop: 6 -2 5 1 5 3 5 6 7 7 8 8 59
7th loop: 5 -2 5 1 5 3 6 6 7 7 8 8 59
8th loop: 5 -2 5 1 3 5 6 6 7 7 8 8 59
9th loop: 5 -2 3 1 5 5 6 6 7 7 8 8 59
10th loop: 1 -2 3 5 5 5 6 6 7 7 8 8 59
11th loop: 1 -2 3 5 5 5 6 6 7 7 8 8 59
12th loop: -2 1 3 5 5 5 6 6 7 7 8 8 59
```

3. Assignment 3

```
.data
   A: .word  7, -2, 5, 1, 129, 192, 12378, 5, 6, 7, 3, 6, 8, 8, 59, 5,
17
   Aend: .word
   nl: .asciz "\n"
   space: .asciz " "
   msg: .asciz "th loop: "

.text
main:
   la a1, A
   la a2, Aend
   addi a2, a2, -4

   j bubble_sort

after_sort:
   li a7, 10
   ecall

bubble_sort:
   add a3, a1, zero      # a3 = &(A[0])

outer:
   addi a3, a3, 4      # a3 = &(A[i+1])
   add a4, a2, zero    # a4 = &(A[n-1])
   li t3, 0            # t3 = swapped = false

   beq a3, a2, after_sort   # a3 == Aend-4?
```

```
inner_loop:
   lw t0, 0(a4)          # t0 = A[j]
   lw t1, -4(a4)          # t1 = A[j-1]

   blt t0, t1, swap      # A[j] < A[j-1] then swap

no_swap:
   addi a4, a4, -4       # j--
   bge a4, a3, inner_loop  # j >= i+1 then continue innerloop

   beqz t3, after_sort   # swapped = false
   j print_array

swap:
   xor t0, t1, t0
   xor t1, t1, t0
   xor t0, t1, t0
   sw t0, 0(a4)
   sw t1, -4(a4)
   li t3, 1             # swapped = true
   j no_swap

print_array:
   la s0, A
   la s1, Aend
   addi s1, s1, -4

   mv a0, a5
   li a7, 1
   ecall
```

```
    addi a5, a5, 1

    la a0, msg   # "nth loop: "
    li a7, 4
    ecall

print_loop:
    lw a0, 0(s0)
    li a7, 1
    ecall

    la a0, space
    li a7, 4
    ecall

    addi s0, s0, 4
    ble s0, s1, print_loop

print_last:
    la a0, nl
    li a7, 4
    ecall

    j outer  # continue sorting
```

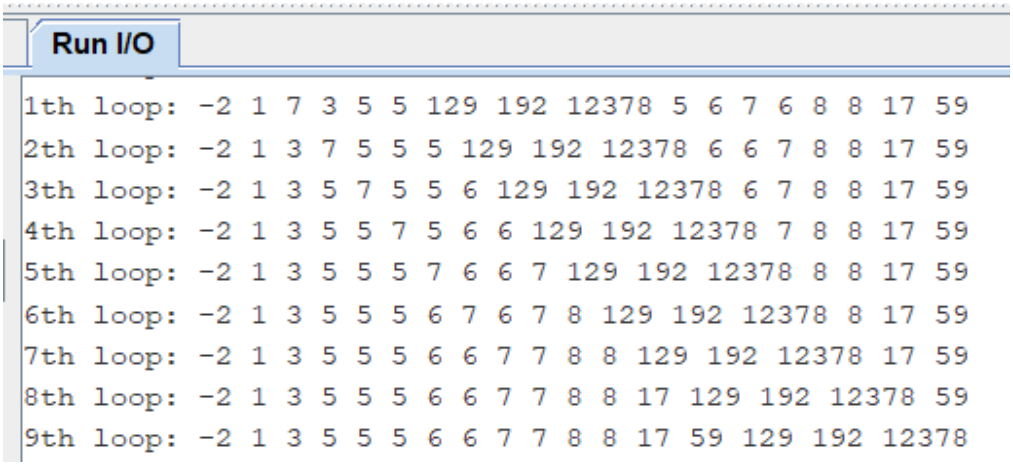
Mảng ban đầu: (từ 0x10010000 đến 0x10010040)

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	7	-2	5	1	129	192	12378	5
0x10010020	6	7	3	6	8	8	59	5
0x10010040	17	2097162	1814063220	980447087	32	0	0	0

Mảng sau khi sort:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	-2	1	3	5	5	5	6	6
0x10010020	7	7	8	8	17	59	129	192
0x10010040	12378	2097162	1814063220	980447087	32	0	0	0

Kết quả in ra:



- Khởi tạo:
  - a1 trỏ vào đầu mảng A.
  - a2 trỏ vào cuối mảng A.
- Vòng lặp ngoài (outer):
  - Duyệt từ A[0] đến A[n-1].
  - Đặt swapped = false (t3 = 0).
  - Vòng lặp trong (inner_loop):
    - Duyệt từ cuối mảng về đầu, so sánh từng cặp phần tử A[j] và A[j-1].
    - Nếu A[j] < A[j-1], thực hiện hoán đổi bằng toán tử XOR. Và đánh dấu swapped = true.
  - Nếu không có hoán đổi (swapped = false), thoát vòng lặp (after_sort).
  - In mảng ra sau mỗi lần loop.
  - Lặp lại cho đến khi mảng được sắp xếp.

4. Assignment 4

```
.data
   A: .word  7, -2, 5, 1, 129, 192, 12378, 5, 6, 7, 3, 6, 8, 8, 59, 5,
17
   Aend: .word
   nl: .asciz "\n"
   space: .asciz " "
   msg: .asciz "th loop: "

.text
main:
   la a1, A
   la a2, Aend
   addi a2, a2, -4
   addi a5, a5, 1       #loop count
   j insertion_sort

after_sort:
   li a7, 10
   ecall

insertion_sort:
   add a3, a1, zero    # &(A[0])
   addi a3, a3, 4      # &(A[1])

outer:
   bgt a3, a2, after_sort   # reached Aend
   lw t0, 0(a3)             # t0 = A[i]
   addi a4, a3, -4          # a4 = &(A[i-1]) = &(A[j])

inner_loop:
   blt a4, a1, insert_key   # checked A[i-1] through A[0]
   lw t1, 0(a4)
   ble t1, t0, insert_key
   sw t1, 4(a4)
   addi a4, a4, -4          # j--
   j inner_loop

insert_key:
   sw t0, 4(a4)
   j print_array

print_array:
   la s0, A
   la s1, Aend
   addi s1, s1, -4
```

```
    addi a0, a5, 0
    li a7, 1
    ecall
    addi a5, a5, 1

    la a0, msg   # "nth loop: "
    li a7, 4
    ecall

print_loop:
    lw a0, 0(s0)
    li a7, 1
    ecall

    la a0, space
    li a7, 4
    ecall

    addi s0, s0, 4
    ble s0, s1, print_loop

print_last:
    la a0, nl
    li a7, 4
    ecall

    addi a3, a3, 4  # i++
    j outer
```

❖ Khởi tạo:
- a1 trỏ vào đầu mảng A.
- a2 trỏ vào cuối mảng A.
- a5 đếm số vòng lặp.

❖ Vòng lặp ngoài (outer):
- Duyệt từ A[1] đến A[n-1].
- Vòng lặp trong (inner_loop):
  - Dịch các phần tử lớn hơn A[i] sang phải.
  - Nếu tìm được vị trí thích hợp, chèn A[i] vào (insert_key).
- In mảng ra sau mỗi lần loop.
- Lặp lại cho đến khi mảng được sắp xếp.

Kết quả in ra:

Run I/O

```
1th loop: -2 7 5 1 129 192 12378 5 6 7 3 6 8 8 59 5 17
2th loop: -2 5 7 1 129 192 12378 5 6 7 3 6 8 8 59 5 17
3th loop: -2 1 5 7 129 192 12378 5 6 7 3 6 8 8 59 5 17
4th loop: -2 1 5 7 129 192 12378 5 6 7 3 6 8 8 59 5 17
5th loop: -2 1 5 7 129 192 12378 5 6 7 3 6 8 8 59 5 17
6th loop: -2 1 5 7 129 192 12378 5 6 7 3 6 8 8 59 5 17
7th loop: -2 1 5 5 7 129 192 12378 6 7 3 6 8 8 59 5 17
8th loop: -2 1 5 5 6 7 129 192 12378 7 3 6 8 8 59 5 17
9th loop: -2 1 5 5 6 7 7 129 192 12378 3 6 8 8 59 5 17
10th loop: -2 1 3 5 5 6 7 7 129 192 12378 6 8 8 59 5 17
11th loop: -2 1 3 5 5 6 6 7 7 129 192 12378 8 8 59 5 17
12th loop: -2 1 3 5 5 6 6 7 7 8 129 192 12378 8 59 5 17
13th loop: -2 1 3 5 5 6 6 7 7 8 8 129 192 12378 59 5 17
14th loop: -2 1 3 5 5 6 6 7 7 8 8 59 129 192 12378 5 17
15th loop: -2 1 3 5 5 5 6 6 7 7 8 8 59 129 192 12378 17
16th loop: -2 1 3 5 5 5 6 6 7 7 8 8 17 59 129 192 12378
```

Chuỗi trước khi sort: (từ 0x10010000 đến 0x10010040)

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	7	-2	5	1	129	192	12378	5
0x10010020	6	7	3	6	8	8	59	5
0x10010040	17	2097162	1814063220	980447087	32	0	0	0

Chuỗi sau khi sort:

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	-2	1	3	5	5	5	6	6
0x10010020	7	7	8	8	17	59	129	192
0x10010040	12378	2097162	1814063220	980447087	32	0	0	0