BÁO CÁO MÔN THỰC HÀNH KIẾN TRÚC MÁY TÍNH

LAB_6: Array and Pointer

Họ và tên: Vũ Thị Quỳnh Như

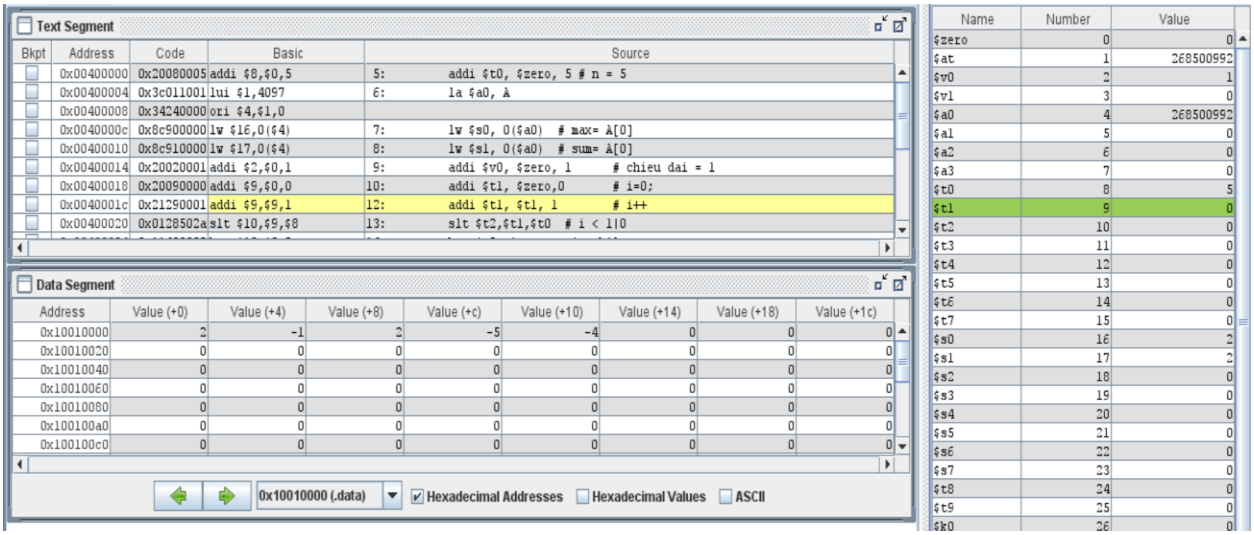
MSSV: 20215110

1. Assignment 1: Tìm tiền tố có tổng lớn nhất

- Code

```
1  # Vu Thi Quynh Nhu
2  .data
3          A: .word 2 -1 2 -5 -4
4  .text
5          addi $t0, $zero, 5 # n = 5
6          la $a0, A
7          lw $s0, 0($a0)  # max= A[0]
8          lw $s1, 0($a0)  # sum= A[0]
9          addi $v0, $zero, 1      # chieu dai = 1
10         addi $t1, $zero,0       # i=0;
11 while:
12         addi $t1, $t1, 1        # i++
13         slt $t2,$t1,$t0  # i < 1|0
14         beq $t2, $zero, exit_while
15         addi $a0, $a0, 4
16         lw $s3, 0($a0)
17         add $s1, $s1,$s3         # sum = sum +A[i]
18         slt $s4, $s0, $s1       # max < sum 1|0
19         beq $s4, $zero, while
20         move $s0, $s1   # max= sum
21         move $v0, $t1   # chieu dai = i
22         j while
23 exit_while:
24         addi $v0, $v0, 1
25
```

- Giải thích
  - Dòng 2-3: Khai báo mảng A
  - Dòng 5-10: Khởi tạo các giá trị ban đầu
    $t0 (n) =5 : số phần tử của mảng
    $a0: Lấy địa chỉ của mảng A
    $s0(max) =A[0]: Tổng tiền tố max ban đầu bằng A[0]
    $s1(sum) =A[0] :Tổng tiền tố ban đầu bằng A[0]
    $v0 (chiều dài) =1: Chiều dài của tiền tố ban đầu bằng 1 (Do đã có A[0])
    $t1(i) =0: Giá trị i=0



- Dòng 11-24: Thực hiện tìm tiền tố có tổng lớn nhất
  - Dòng 12: i=i+1
  - Dòng 13: Thực hiện so sánh nếu i < n thì $t2=1 ngược lại $t2=0
  - Dòng 14: Nếu $t2=0 (i>=n) thì chuyển tới exit_while
  - Dòng 15-16: Thực hiện lấy ra A[i] gán vào $s3
  - Dòng 17: Thực hiện tính sum=sum+A[i]
  - Dòng 18: Thực hiện so sánh nếu max < sum thì $s4=1 ngược lại $s4=0
  - Dòng 19: Nếu $s4=0 (max>=sum) thì quay lại while (dòng 11) để tiếp tục vòng lặp. Nếu $s4 !=) thì thực hiện các dòng dưới
  - Dòng 20: Gán giá trị max=sum
  - Dòng 21: Gán chiều dài tiền tố = i
  - Dòng 22: Nhảy tới while để tiếp tục vòng lặp
  - Dòng 24: ($v0=$v0+1) là chiều dài tiền tố cần tìm do i bắt đầu từ 0

Kết quả cần tìm sẽ là max ($s0) và chiều dài tiền tố ($v0)

Và với A=(2,-1,2,-5,-4) thì max=2+ (-1) +2 =3 và chiều dài tiền tố là 3

$v0	2	3
$v1	3	0
$a0	4	268501008
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	5
$t1	9	5
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	3

2. Assignment 2: Thực hiện sắp xếp tăng dần (theo selection sort)

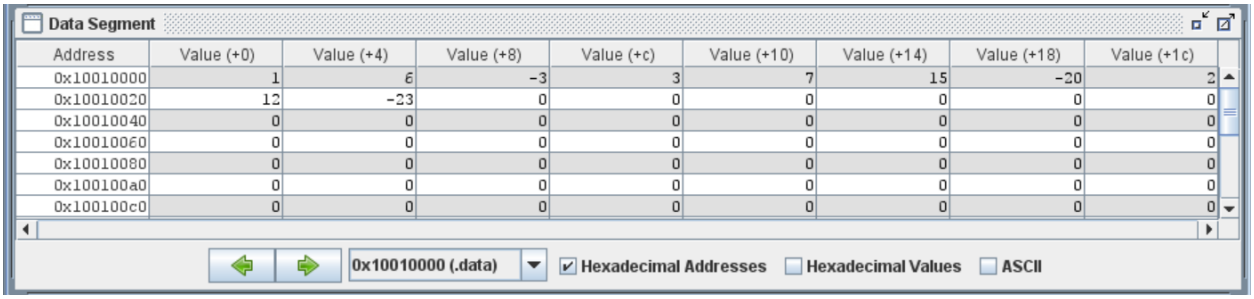
- Code

```
1  # Vu Thi Quynh Nhu
2  .data
3          A: .word 1 6 -3 3 7 15 -20 2 12 -23
4  .text
5  set:
6          addi $t0, $zero, 10 # n
7          addi $t1, $zero, 9 # i
8          la $a0, A
9  end_set:
10 sort:
11         for1:
12                 slt $t2, $t1, $zero # i < 0 1|0
13                 bne $t2, $zero,end_sort
14                 move $v0, $zero # vi tri maxdoan=0
15                 lw $s0, 0($a0) # maxdoan = A[0]
16                 move $t3, $zero # j=0
17                 for2:
18                         slt $t2, $t1, $t3 # i < j 1|0
19                         bne $t2, $zero, end_for2
20                         max:
21                                 sll $t4, $t3 ,2
22                                 add $a1, $a0, $t4
23                                 lw $s1, 0($a1) # A[j]
24                                 slt $t2, $s0, $s1 # maxdoan < A[j] 1|0
25                                 beq $t2, $zero, end_max
26                                 move $v0, $t3 # vi tri maxdoan=j
27                                 move $s0, $s1 # maxdoan=A[j]
28                         end_max:
29                                 addi $t3, $t3, 1 # j++
30                                 j for2
31                 end_for2:
32                 j doicho
33                 doicho:
34                         sll $t4, $t1,2
35                         add $a1, $a0, $t4
36                         lw $s3, 0($a1) # tam=A[i]
37                         sll $t4, $v0, 2
38                         add $a2, $a0, $t4
39                         lw $s4, 0($a2) # $s4= A[maxdoan]
40                         sw $s4, 0($a1) # A[i] = A[maxdoan]
41                         sw $s3, 0($a2) # A[maxdoan]=tam
42                         j end_doicho
43                 end_doicho:
44                 sub $t1, $t1, 1
45                 j for1
46         end_for1:
47 end_sort:
```

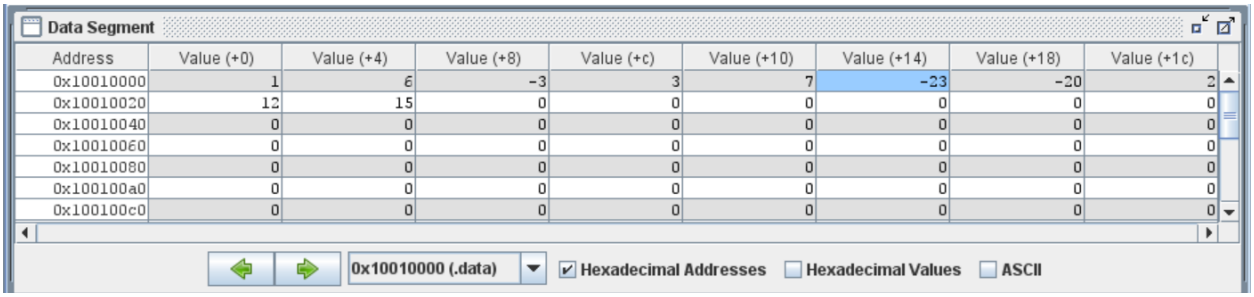
- Giải thích
  - Dòng 3: Khai báo mảng A
  - Dòng 5-9: Set các giá trị ban đầu n($t0)=10, i($t1)=9, địa chỉ mảng A ($a0)
  - Dòng 10-47: Thực hiện sắp xếp
    - Dòng 12-13: Nếu i < 0 thì chuyển tới end_sort ngược lại thì thực hiện các câu lệnh sau
    - Dòng 15-16:Gán maxdoan =0 và j=0

- Dòng 18-19: Nếu i<j thì chuyển đến end_for2 ngược lại thực hiện các câu lệnh dưới
- Dòng 20-29: Thực hiện so sánh nếu maxdoan<A[j] thì maxdoan=A[j] và vị trí maxdoan=j ngược lại thì j=j+1
- Dòng 30: Nhảy tới for2: tiếp tục vòng lặp for2
- Dòng 32-43: Thực hiện đổi chỗ 2 phần tử maxdoan và A[i]
- Dòng 44: i=i-1
- Dòng 45: nhảy tới for1: tiếp tục vòng lặp for1

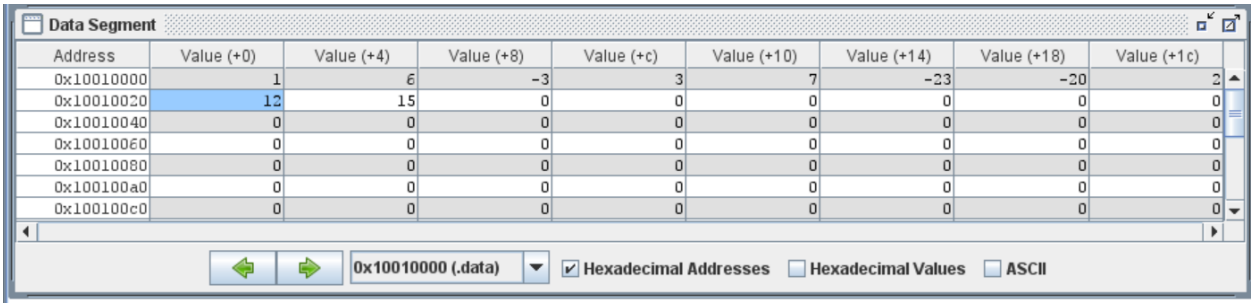
- Kết quả chạy
  Mảng ban đầu:



Khi i=9



Khi i=8



Khi i=7

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1	6	-3	3	2	-23	-20	7
0x10010020	12	15	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

Khi i=6

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1	-20	-3	3	2	-23	6	7
0x10010020	12	15	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

Khi i=5

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1	-20	-3	-23	2	3	6	7
0x10010020	12	15	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

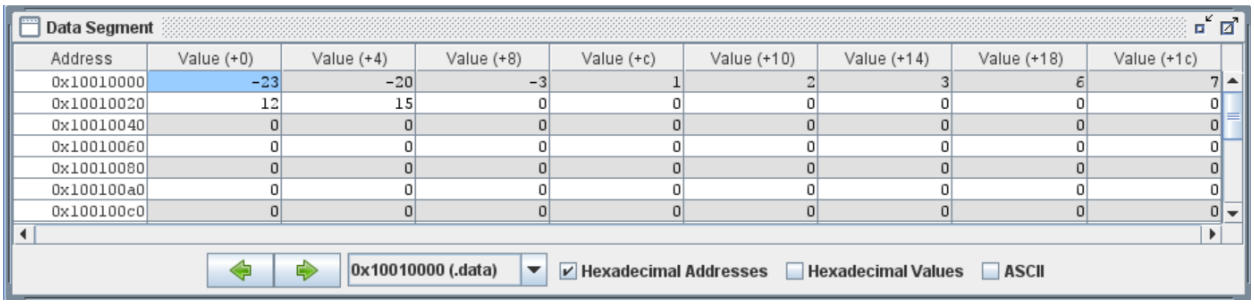
Khi i=4

Data Segment

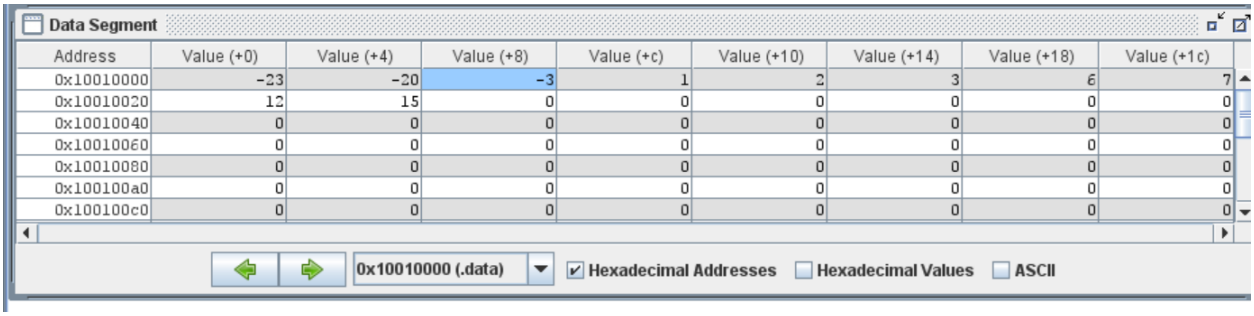
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1	-20	-3	-23	2	3	6	7
0x10010020	12	15	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

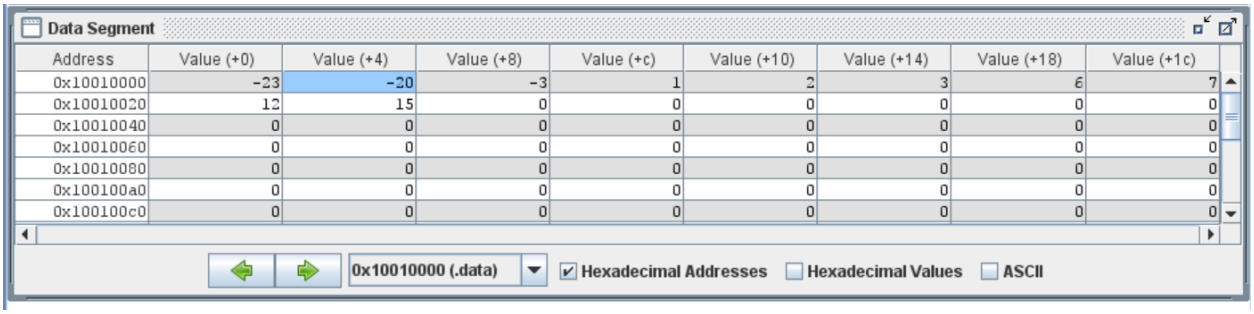
Khi i=3



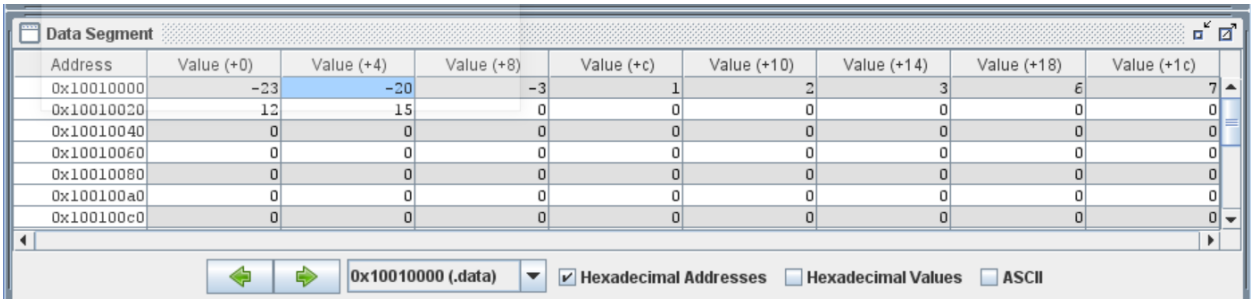
Khi i=2



Khi i=1



Khi i=0 (Kết thúc sắp xếp)



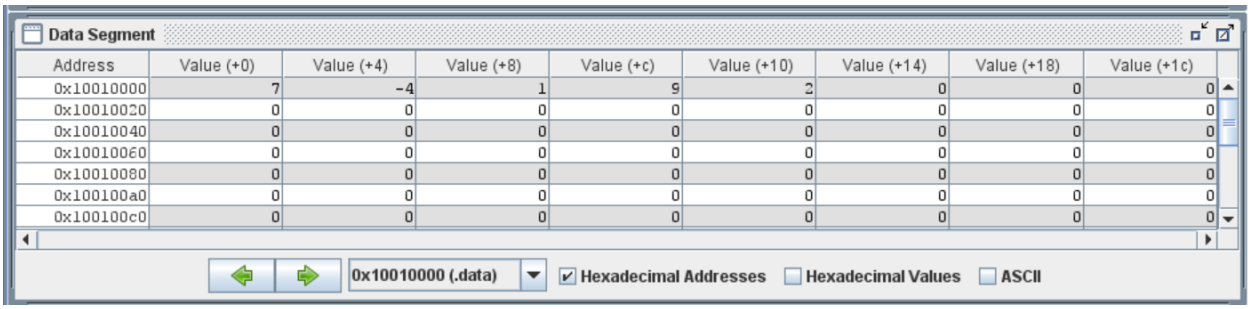
3. **Assignment 3: Thực hiện sắp xếp (Theo Bubble sort)**
   Trường hợp 1: Sắp xếp tăng dần
   - Code

```
1  # Vu Thi Quynh Nhu
2  .data
3          A: .word 7 -4 1 9 2
4  .text
5  set:
6          addi $t0, $zero, 5 # n=5
7          addi $t1, $zero, 4 # i=4
8          la $a0, A
9  end_set:
10 sort:
11         for1:
12                 slt $t2, $t1, $zero # i < 0 1|0
13                 bne $t2, $zero,end_sort
14                 addi $t3, $zero, 1 #j=1
15                 for2:
16                         slt $t2, $t1, $t3 # i < j 1|0
17                         bne $t2, $zero, end_for2
18                         sosanh:
19                                 sll $t4, $t3 ,2
20                                 add $a1, $a0, $t4
21                                 lw $s1, 0($a1) # A[j]
22                                 sub $a2, $a1, 4
23                                 lw $s0, 0($a2) # A[j-1]
24                                 slt $t2, $s0, $s1 # A[j-1] < A[j] 1|0
25                                 bne $t2, $zero, end_sosanh
26                                 doicho:
27                                         lw $s3, 0($a2) # tam=A[j-1]
28                                         lw $s4, 0($a1) # $s4= A[j]
29                                         sw $s4, 0($a2) # A[j-1]=A[j]
30                                         sw $s3, 0($a1) # A[j]=tam
31                                         j end_doicho
32                                 end_doicho:
33                         end_sosanh:
34                                 addi $t3, $t3, 1 # j++
35                                 j for2
36                 end_for2:
37                 sub $t1, $t1, 1
38                 j for1
39         end_for1:
40 end_sort:
```

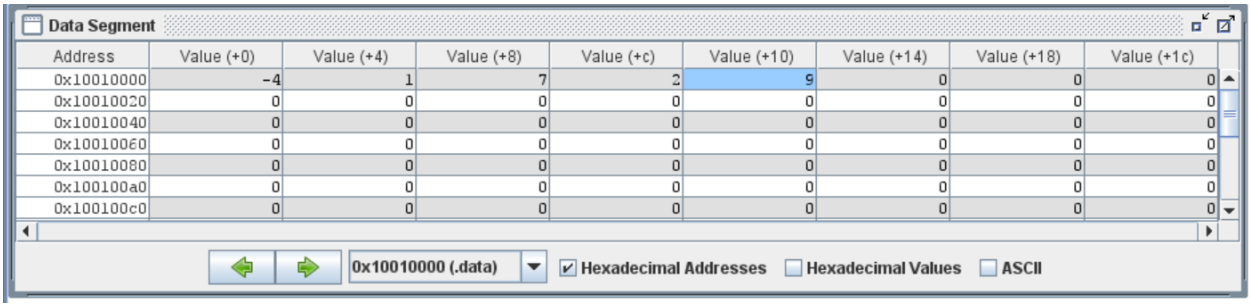
- Giải thích
  - Dòng 3: Khai báo mảng A
  - Dòng 5-9: Khởi tạo n ($t0)=5 và i ($t1)=4, địa chỉ mảng A: $a0
  - Dòng 10-40: Thực hiện sắp xếp
    - Dòng 12-13: Nếu i<0 thì chuyển tới end_sort ngược lại thì thực hiện các câu lệnh sau đó
    - Dòng 14: Gán j=1
    - Dòng 16-17: Nếu i<j thì chuyển tới end_for2 ngược lại thì thực hiện các câu lệnh sau đó
    - Dòng 18-35: Thực hiện so sánh nếu A[j]<= A[j-1] thì đổi chỗ A[j] và A[j-1] ngược lại thì j=j+1 và tiếp tục vòng lặp for2

- Dòng 37: i=i-1
- Dòng 38: chuyển tới for1: tiếp tục vòng lặp for 1

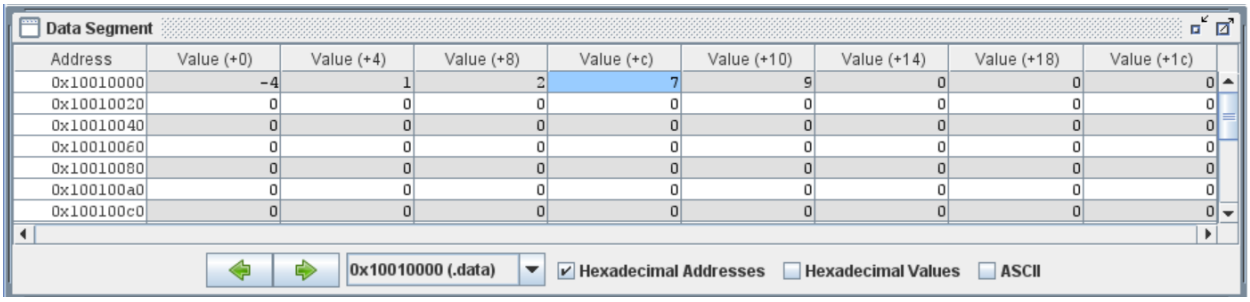
- Kết quả chạy:
  Mảng A ban đầu



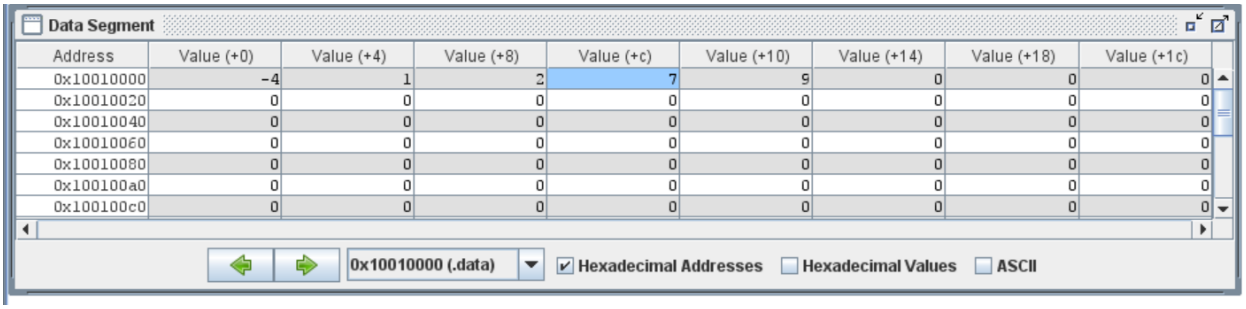
Khi i=4



Khi i=3



Khi i=2



Khi i=1

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	-4	1	2	7	9	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Khi i=0 (Kết thúc sắp xếp)

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	-4	1	2	7	9	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Trường hợp 2: Sắp xếp giảm dần

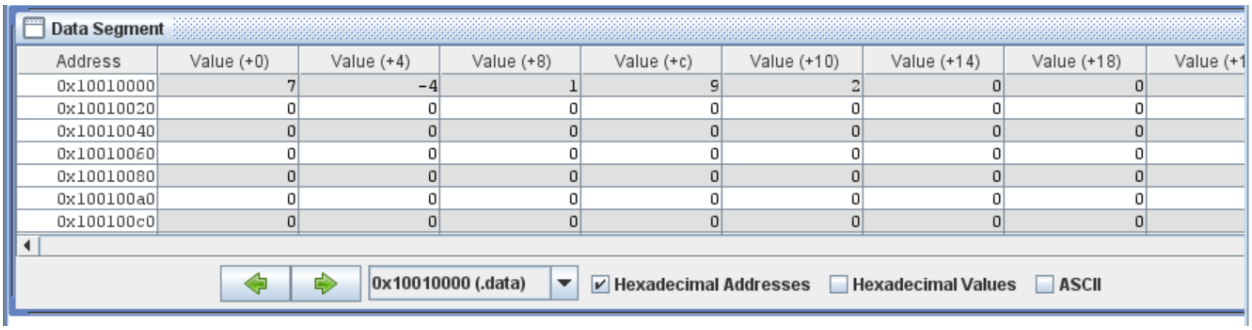
- Code

```
# Vu Thi Quynh Nhu
.data
        A: .word 7 -4 1 9 2
.text
set:
        addi $t0, $zero, 5 # n=5
        addi $t1, $zero, 4 # i=4
        la $a0, A
end_set:
sort:
        for1:
                slt $t2, $t1, $zero # i < 0 1|0
                bne $t2, $zero,end_sort
                addi $t3, $zero, 1 #j=1
                for2:
                        slt $t2, $t1, $t3 # i < j 1|0
                        bne $t2, $zero, end_for2
                        sosanh:
                                sll $t4, $t3 ,2
                                add $a1, $a0, $t4
                                lw $s1, 0($a1) # A[j]
                                sub $a2, $a1, 4
                                lw $s0, 0($a2) # A[j-1]
                                slt $t2, $s1, $s0 # A[j] < A[j-1] 1|0
                                bne $t2, $zero, end sosanh
```

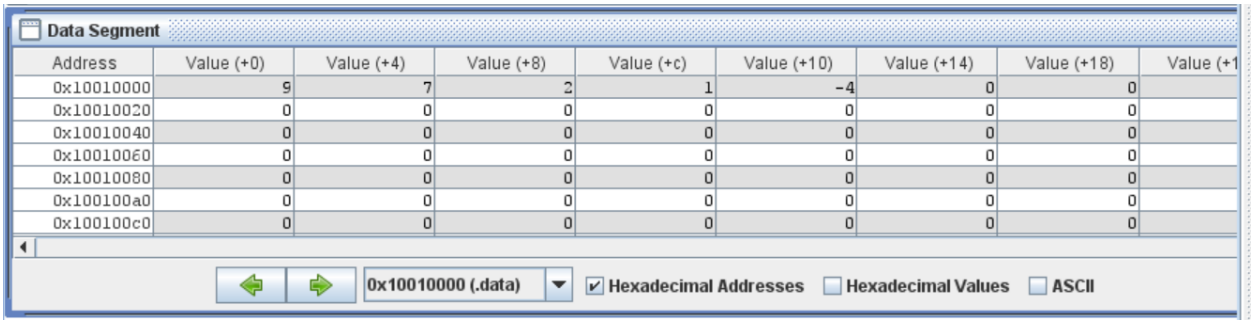
```
26                              doicho:
27                                      lw $s3, 0($a2) # tam=A[j-1]
28                                      lw $s4, 0($a1) # $s4= A[j]
29                                      sw $s4, 0($a2) # A[j-1]=A[j]
30                                      sw $s3, 0($a1) # A[j]=tam
31                                      j end_doicho
32                              end_doicho:
33                      end_sosanh:
34                              addi $t3, $t3, 1 # j++
35                              j for2
36              end_for2:
37              sub $t1, $t1, 1
38              j for1
39      end_for1:
40  end_sort:
```

- Giải thích
    - Tương tự như trường hợp tăng dần nhưng đổi điều kiện:
      Thực hiện so sánh nếu A[j]>= A[j-1] thì đổi chỗ A[j] và A[j-1] ngược lại thì j=j+1 và tiếp tục vòng lặp for2

- Kết quả chạy
  Mảng A ban đầu



Sau khi chạy:



4. Assignment 4: Thực hiện sắp xếp (Theo Insertion sort)
Trường hợp 1: Sắp xếp tăng dần

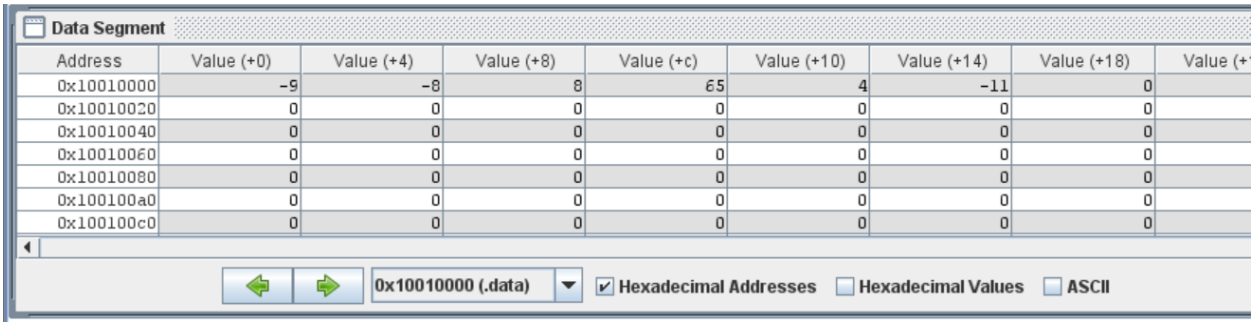
- Code

```
1  # Vu Thi Quynh Nhu
2  .data
3          A: .word -9 -8 8 65 4 -11
4  .text
5          la $a0, A
6          addi $t4, $zero, 6 # n=6
7          addi $t1, $t1,1 # i=1
8          for:
9                  slt $s4, $t1, $t4 # i < n 1|0
10                 beq $s4, $zero,end_for
11                 sub $t2, $t1, 1 # j=i-1
12                 sll $s4, $t1, 2
13                 add $s4, $a0, $s4
14                 lw $s1, 0($s4) # A[i]
15                 while:
16                         slt $s0, $t2, $zero # j<0 1|0
17                         sll $t3, $t2, 2
18                         add $a1, $a0, $t3
19                         lw $s2, 0($a1) # A[j]
20                         slt $s3, $s2, $s1 # A[j] < A[i] 1|0
21                         add $s3, $s3, $s0
22                         bne $s3, $zero,end_while
23                         sw $s2, 4($a1) # A[j+1] = A[j]
24                         sub $t2, $t2, 1
25                         j while
26                 end_while:
27
28                     sw $s1, 4($a1) # A[i]=A[j+1]
29                     addi $t1,$t1,1 # i++
30                     j for
31             end_for:
```

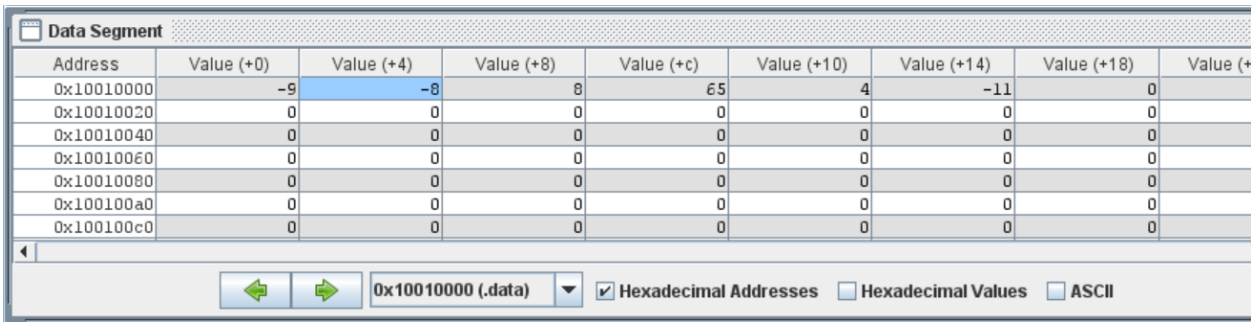
- Giải thích
  - Dòng 3: Khai báo mảng A
  - Dòng 5-7: Khởi tạo các giá trị : địa chỉ mảng A ($a0), n=6, i=1
  - Dòng 8-31: Thực hiện sắp xếp
    - Dòng 9-10: Nếu i<n thì tiếp tục chương trình ngược lại thì chuyển tới end_for
    - Dòng 11: j=i-1
    - Dòng 12-14: Lấy ra A[i]
    - Dòng 16-25: Nếu j>=0 && A[i] <= A[j] thì {A[j+1]=A[j] và j=j-1 sau đó chuyển tới while (tiếp tục vòng lặp while)} ngược lại thì chuyển tới end_while

- Dòng 28: A[j+1]=A[i]
- Dòng 29: i=i+1
- Dòng 30: Chuyển tới for : tiếp tục thực hiện vòng lặp for

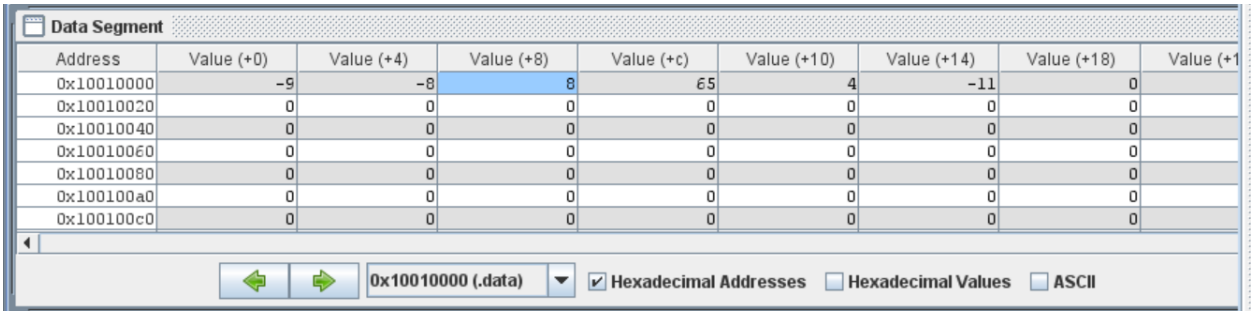
- Kết quả chạy
  Mảng A ban đầu



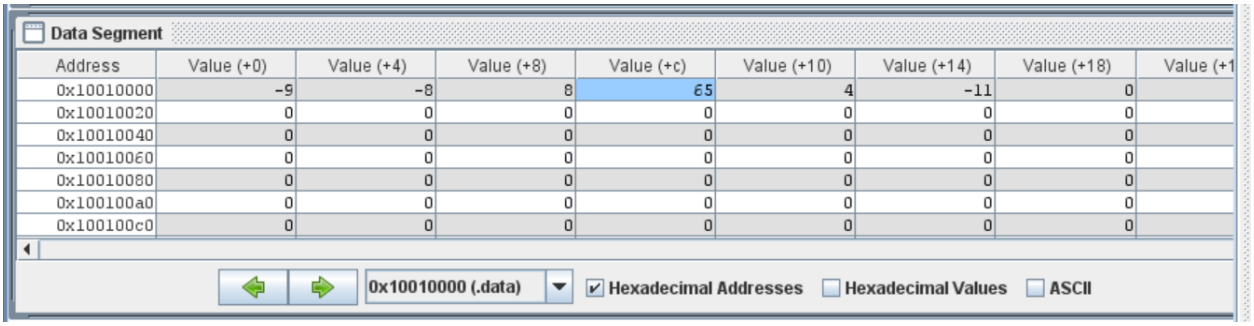
Khi i=1



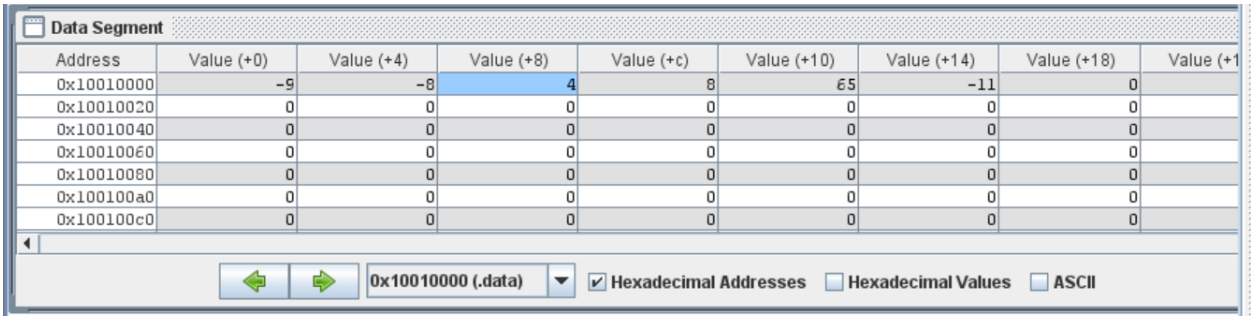
Khi i=2



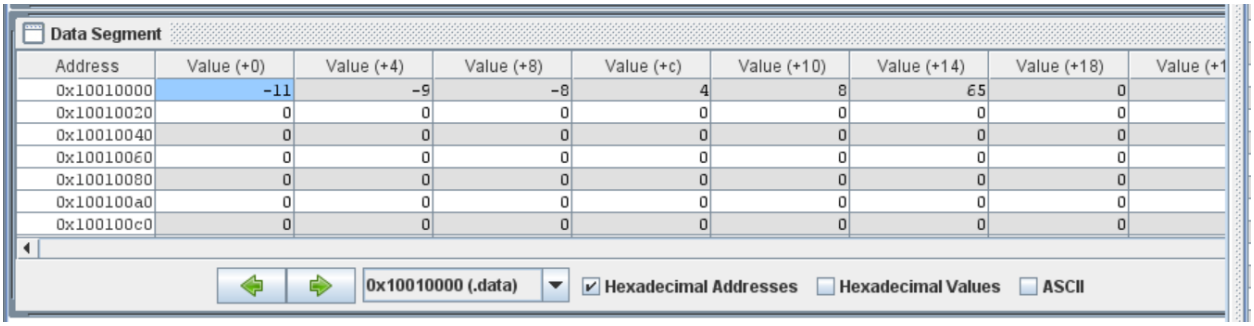
Khi i=3



Khi i=4



Khi i=5 (Kết thúc sắp xếp)



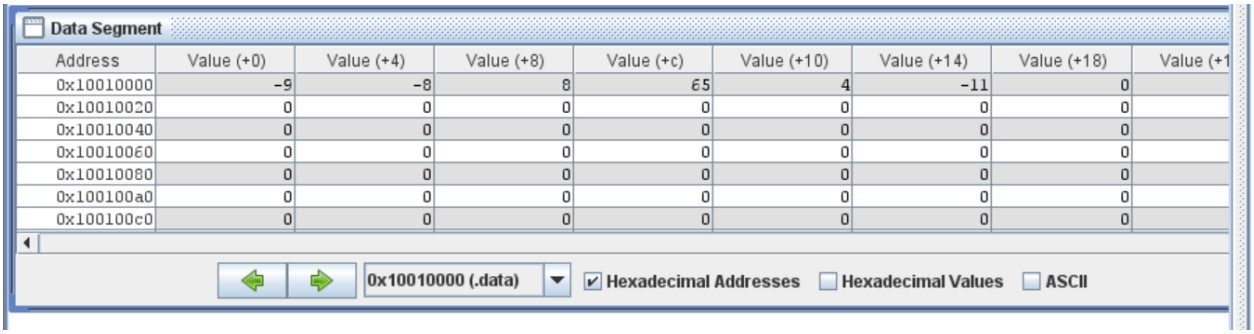
Trường hợp 2: Sắp xếp giảm dần

- Code

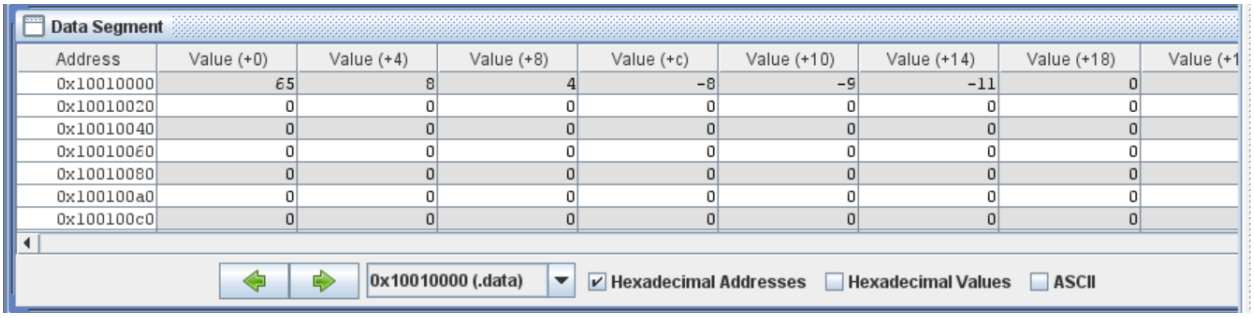
```
1  # Vu Thi Quynh Nhu
2  .data
3          A: .word -9 -8 8 65 4 -11
4  .text
5          la $a0, A
6          addi $t4, $zero, 6 # n=6
7          addi $t1, $t1,1 # i=1
8          for:
9                  slt $s4, $t1, $t4 # i < n 1|0
10                 beq $s4, $zero,end_for
11                 sub $t2, $t1, 1 # j=i-1
12                 sll $s4, $t1, 2
13                 add $s4, $a0, $s4
14                 lw $s1, 0($s4) # A[i]
15                 while:
16                         slt $s0, $t2, $zero # j<0 1|0
17                         sll $t3, $t2, 2
18                         add $a1, $a0, $t3
19                         lw $s2, 0($a1) # A[j]
20                         slt $s3, $s1, $s2 # A[i] < A[j] 1|0
21                         add $s3, $s3, $s0
22                         bne $s3, $zero,end_while
23                         sw $s2, 4($a1) # A[j+1] = A[j]
24                         sub $t2, $t2, 1
25                         j while
26                 end_while:
27
28                 sw $s1, 4($a1) # A[j+1]=A[j]
29                 addi $t1,$t1,1 # i++
30                 j for
31         end_for:
32
```

- Giải thích
  - Tương tự như sắp xếp tăng dần nhưng thay điều kiện:
  - Nếu j>=0 && A[i] >= A[j] thì {A[j+1]=A[j] và j=j-1 sau đó chuyển tới while (tiếp tục vòng lặp while)} ngược lại thì chuyển tới end_while

- Kết quả chạy
  Mảng A ban đầu



Sau khi chạy


Data Segment
Address
Value (+0)
Value (+4)
Value (+8)
Value (+c)
Value (+10)
Value (+14)
Value (+18)
0x10010000 65 8 4 -8 -9 -11 0
0x10010020 0 0 0 0 0 0 0
0x10010040 0 0 0 0 0 0 0
0x10010060 0 0 0 0 0 0 0
0x10010080 0 0 0 0 0 0 0
0x100100a0 0 0 0 0 0 0 0
0x100100c0 0 0 0 0 0 0 0
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII