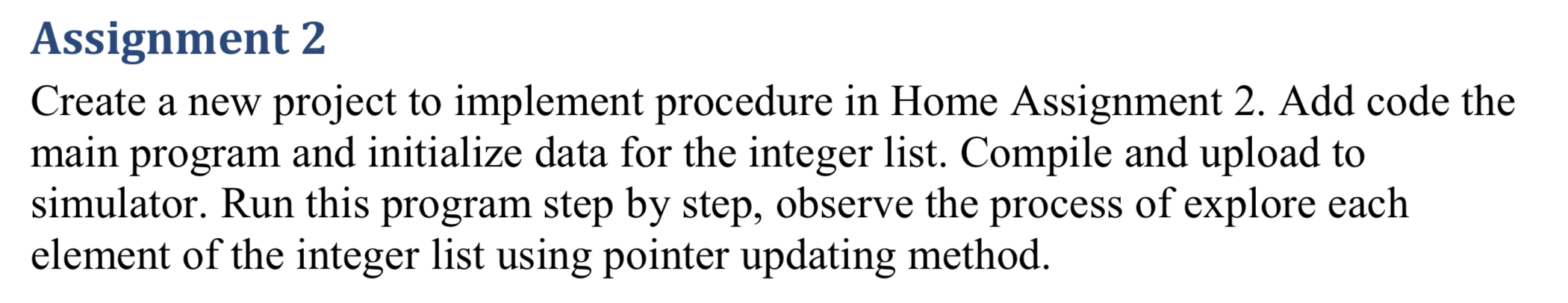
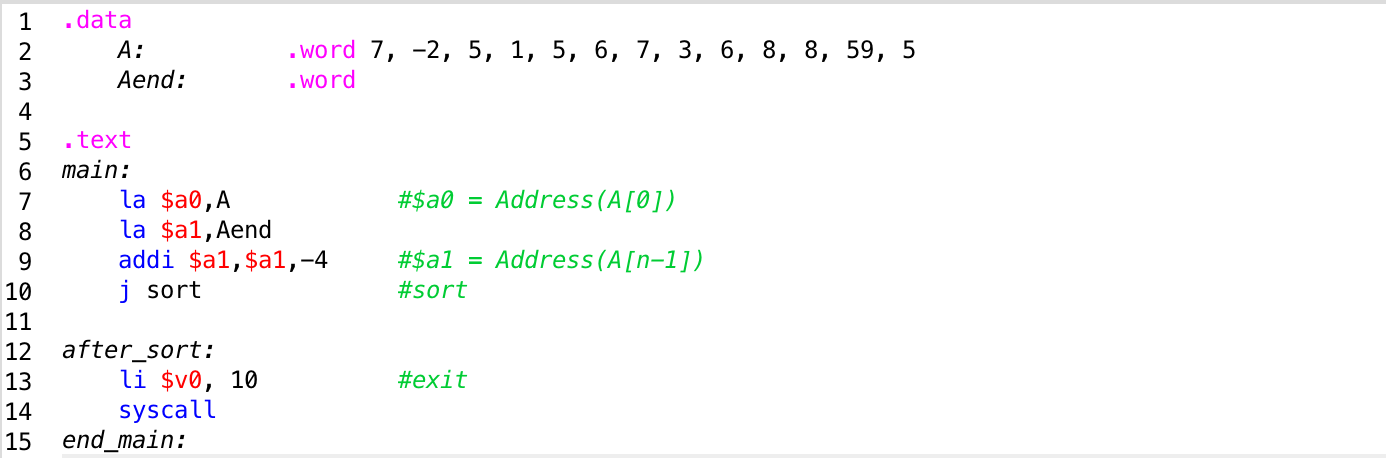
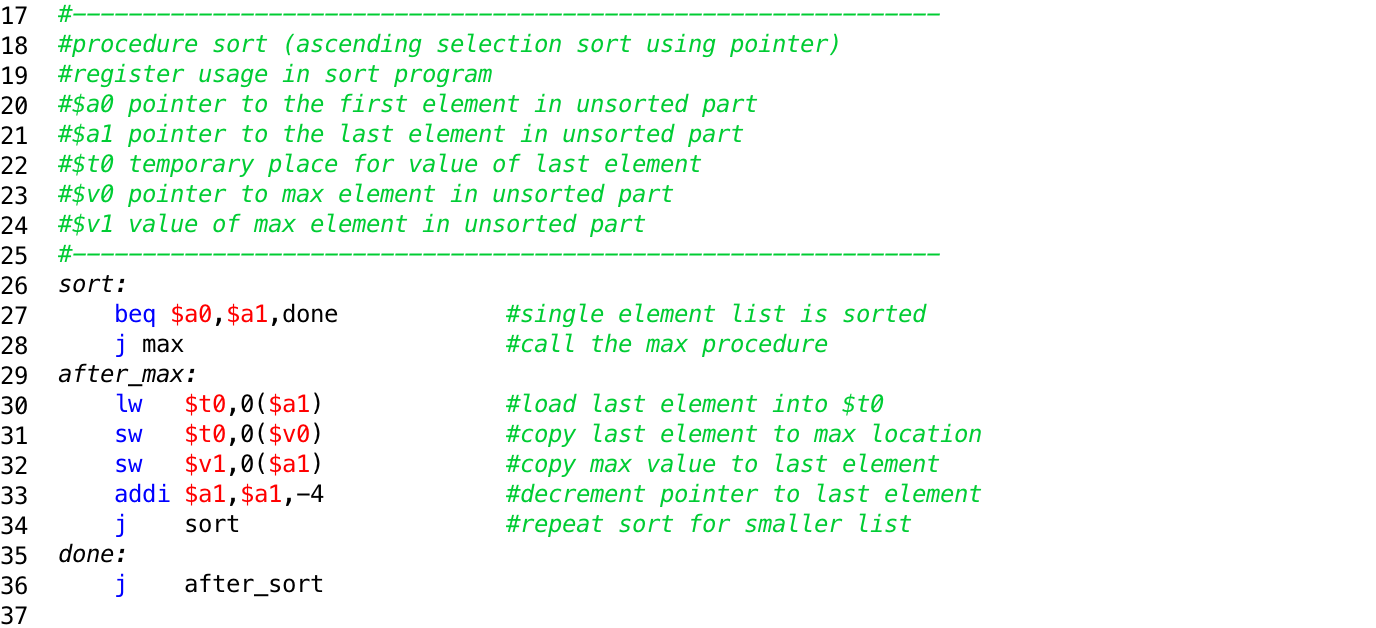
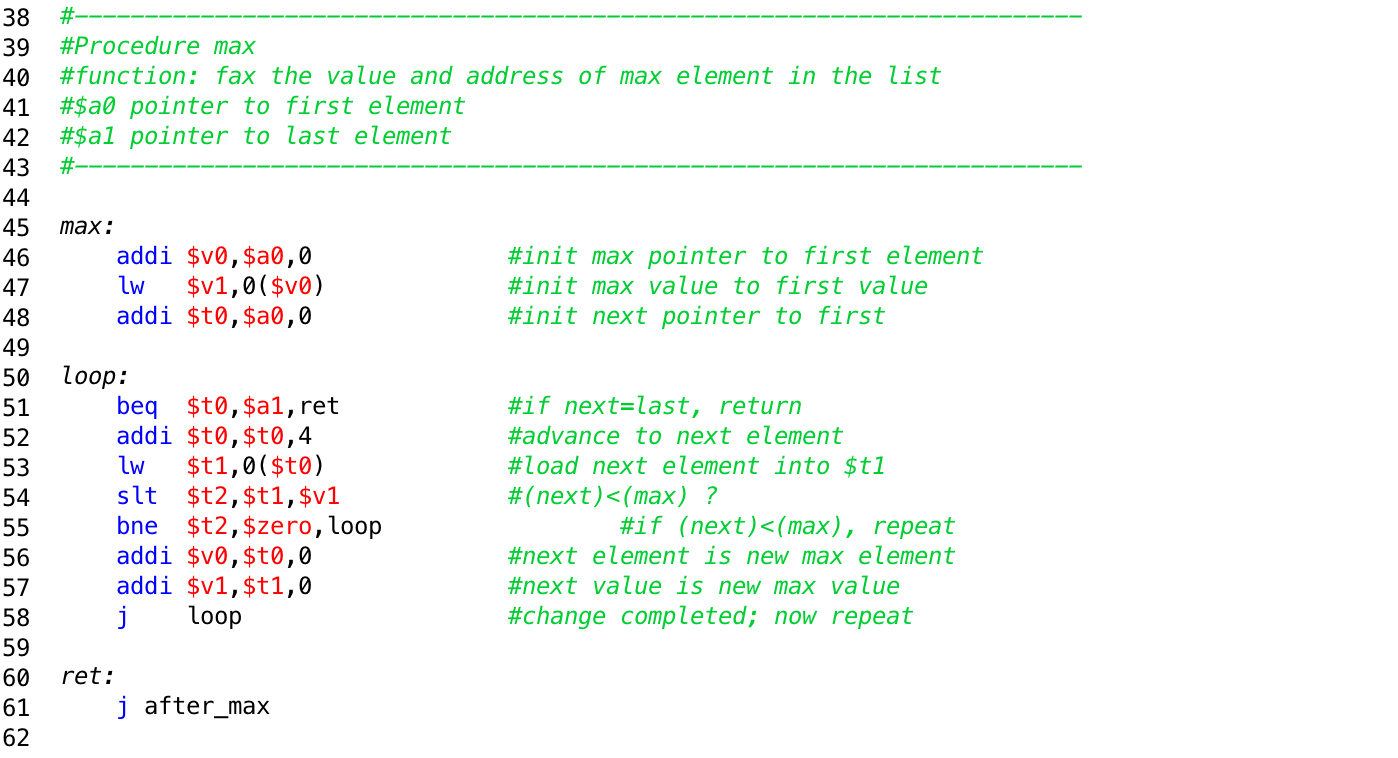
|  |
| --- |
| Nguyễn Trung Thành - 20176874 |



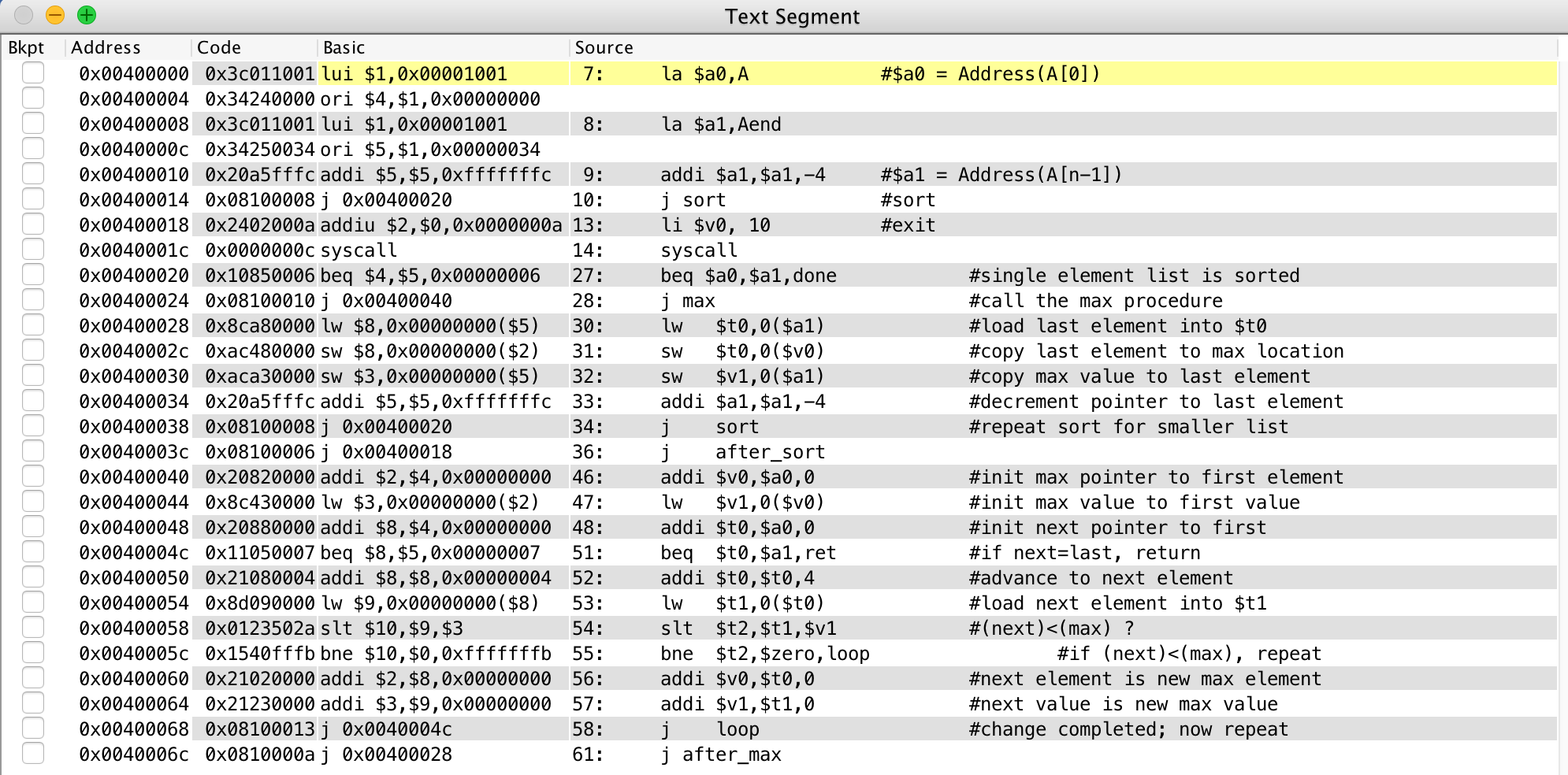
* CODE



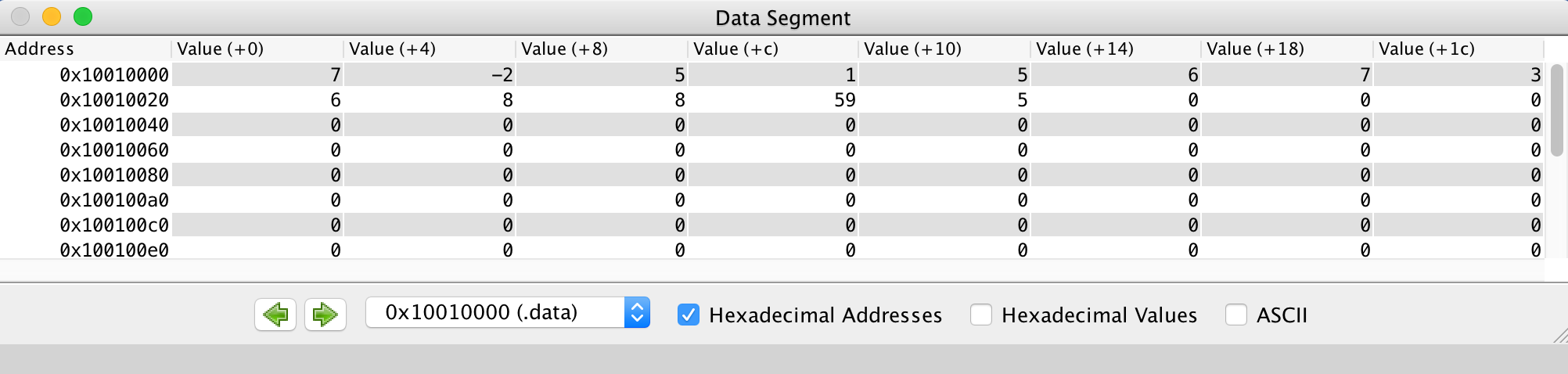




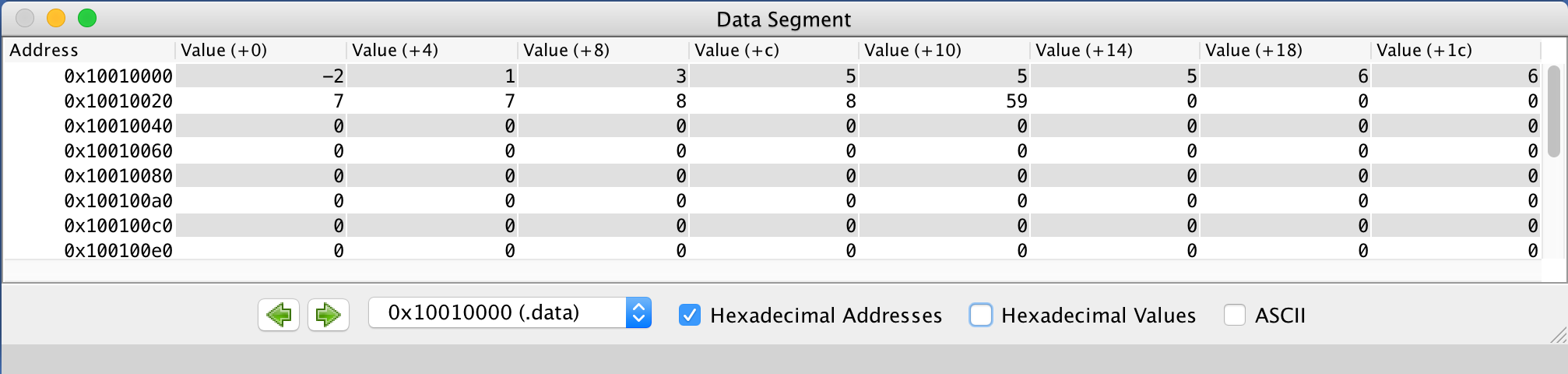
* Bảng Text Segment



* Mảng trước khi sắp xếp



* Mảng sau khi sắp xếp



Giải thích chi tiết:

#--------------------------------------------------------------

sort:

beq $a0,$a1,done #single element list is sorted

j max #call the max procedure

after\_max:

lw $t0,0($a1) #load last element into $t0

sw $t0,0($v0) #copy last element to max location

sw $v1,0($a1) #copy max value to last element

addi $a1,$a1,-4 #decrement pointer to last element

j sort #repeat sort for smaller list

done:

j after\_sort

#--------------------------------------------------------------

1. Đầu tiên kiểm tra con trỏ ở đầu và ở cuối có trùng nhau nhau, nếu trùng nhau => done;

Nếu không thực hiện tìm max trong mảng.

1. Sau khi tìm max xong hoán đổi vị trí phần tử max xuống cuối $v1 => sw vào $a1.
2. Sau đó giảm -4 để lấy phần tử kề phần tử cuối vừa xét.
3. Thực hiện gọi sort để lặp lại quy trình trên.
4. Nếu $a0 trùng với $a1 => kết thúc

* Gọi after\_sort vào kết thúc chương trình.

#--------------------------------------------------------------

* Chạy chương trình:

1. Đầu tiên: $a0: 0x10010000 (vị trí phần tử đầu tiên)

$a1: 0x10010030 (vị trí phần tử cuối cùng)

* Không trùng sau => tìm max và hoán đổi max xuống cuối cùng

1. Thực hiện vòng lặp tìm max:

* Tìm thấy max có giá trị $v1: 0x0000003b (tương đương 59)

1. Sau khi tìm max, hoán đổi max xuống vị trí cuối cùng

Và giảm $a1 xuống phần tử A[n-2] => $a1: 0x1001002c

1. Lặp lại sort: do $a1 không trùng $a0

* Chương trình tiếp tục lặp lại các bước như trên.

....

Thực hiện tương tự cho đến khi giá trị $a0: 0x10010000

và $a1: 0x10010000

* Kết thúc chương trình. Và đã thành công thực hiện sắp xếp theo thứ tự tăng dần.