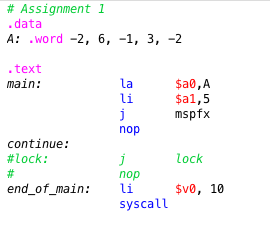
Laboratory Exercise 6

Array and Pointer

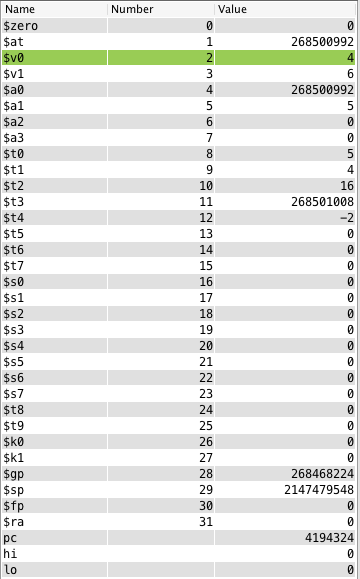
**# Assignment 1:**



* Thêm lệnh syscall 10 để thoát chương trình.

****

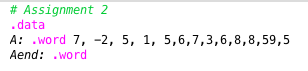
* Lưu ý vì $v0 đang chứa giá trị length của tiền tố có tổng lớn nhất trong dãy. Nên ta sẽ chạy hết chương trình và lùi lại 1 lệnh để quan sát được kết quả.

****

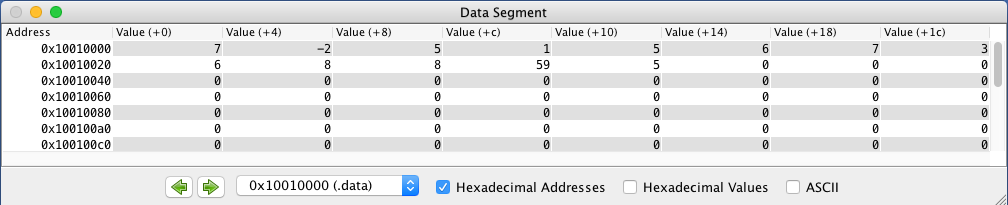
Độ dài của tiền tố có tổng lớn nhất là: $v0 = 4 (tức là 4 phần tử đầu tiên)

Tổng của tiền tố này là : $v1 = 6 ( = -2 + 6 – 1 + 3)

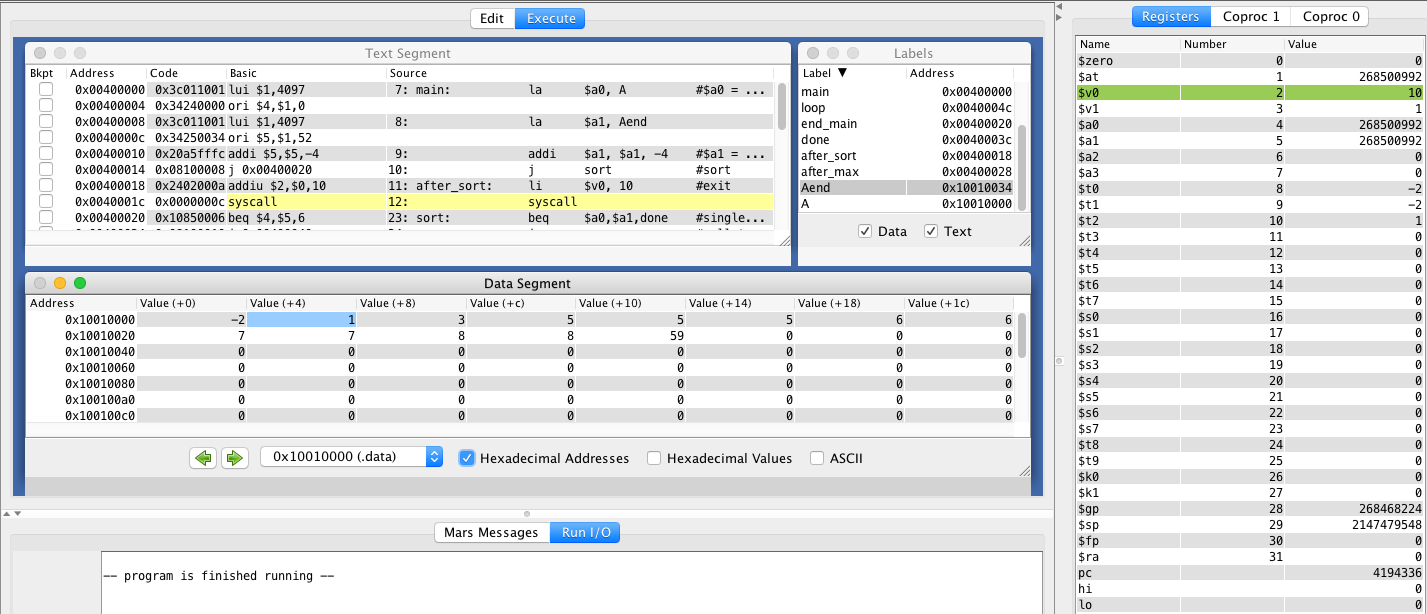
**# Assignment 2:**



* Trước khi sort:



* Sau khi sort:



**# Assignment 3:**

* Pseudocode: **Mảng A có n phần tử:**

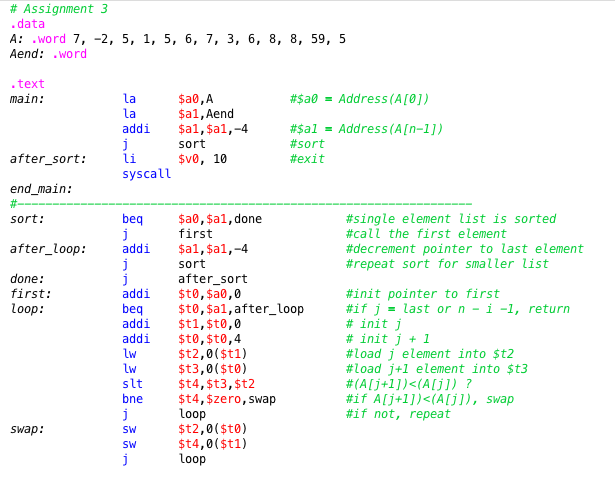
**Def BubbleSort(int A{}, int n):**

For i in range(0,n-1):

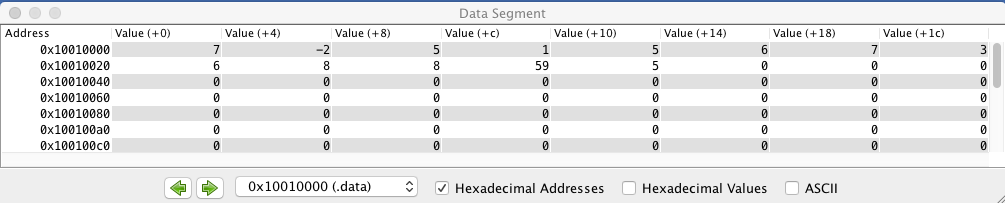
For j in range(0, n-i-1):

If A[j] > A[j+1]: swap(&A[j], &A[j+1])

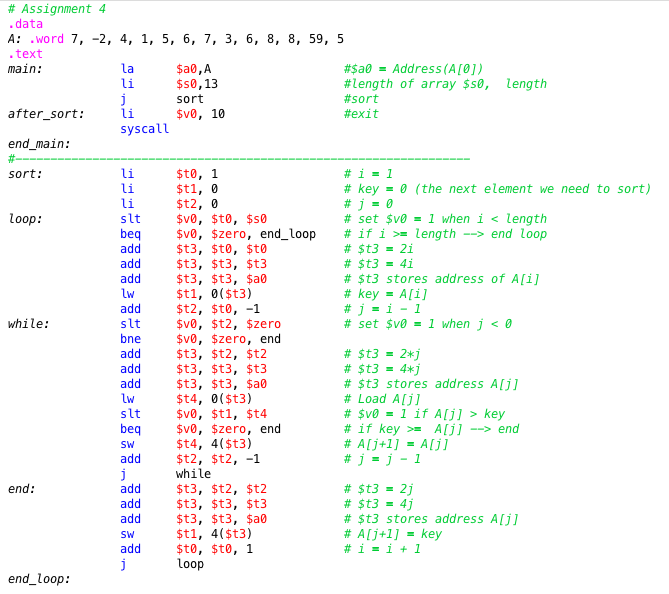
* Trong vòng lặp j, có sự so sánh giữa phần tử ở con trỏ hiện tại ($t2) và phần tử ở con trỏ next ($t3), nếu phần tử hiện tại lớn hơn thì sẽ swap 2 vị trí này (bubble)
* Sau mỗi vòng lặp của loop, size của array sẽ giảm đi 1 (loại đi phần tử max vừa tìm đc ở loop j)

****

* **Kết quả:**

****

**# Assignment 4:**

****

* **Pseudocode:**

**Def InsertionSort(int A{}, int n):**

For i in range(1, n):

key = arr[i]

j = i – 1

While (j>=0 && arr[j] > key):

arr[j+1] = arr[j]

j = j – 1

arr[j+1] = key

* Key sẽ chọn ra phần tử đầu tiên của dãy chưa được sắp xếp.
* Vòng lặp while cho dịch tất cả các phần tử của dãy đã sắp xếp sang phải để ra vị trí trống cho phần tử key chèn vào.

**# Conclusion**

* **Indexing:**
* Advantage: Trực quan, dễ hiểu, dễ khởi tạo.
* Disadvantage: Mất thời gian để tính địa chỉ.
* **Pointer:**
  + Advantage: Nhanh, ít tốn bộ nhớ, không cần tính toán.
  + Disadvantage: Khó hiểu, nhiều khi mất kiểm soát, rối.