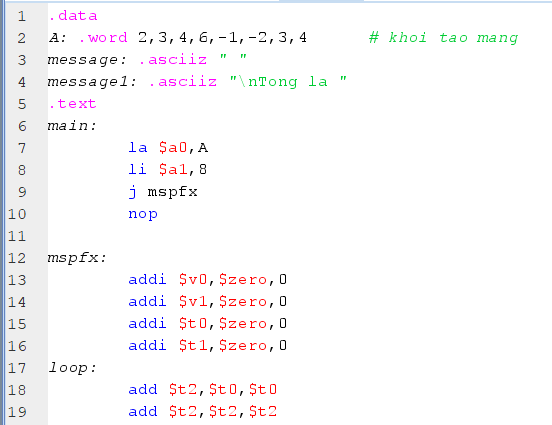
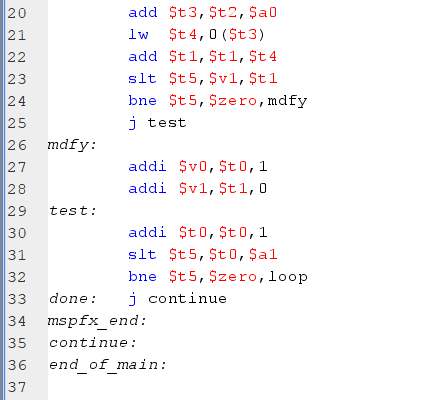
**Bài thực hành số 6**

**Lớp : 139365 – Học phần: Thực hành Kiến trúc máy tính**

**Họ và tên : Nguyễn Thị Thùy Dung MSSV : 20215009**

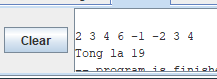
**Bài 1:**





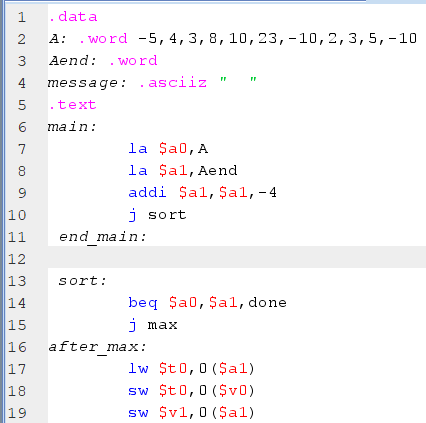
*Thực hiện gõ chương trình với công cụ MARS*

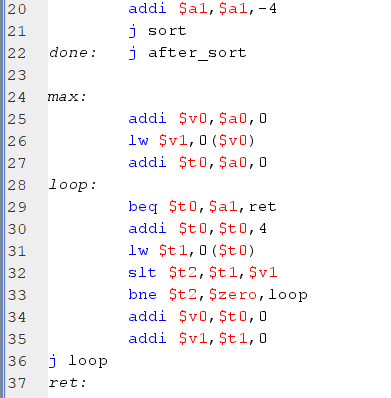
* Kết quả :

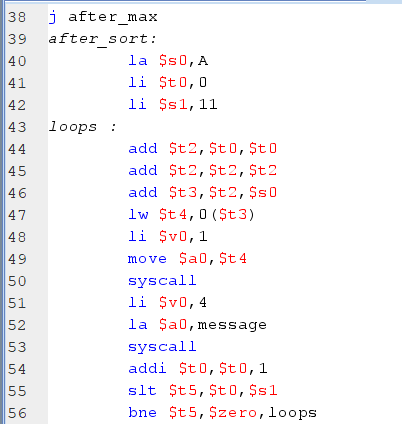


* Giải thích :
* Khai báo (dòng 2-4): Khai báo mảng A có 8 phần tử
* Dòng 7-8 : lấy giá trị phần tử đầu tiên của mảng A, khởi tạo n = 8 (số phần tử của mảng bằng 8).
* Khởi tạo (dòng 13-16) : khởi tạo độ dài chuỗi tiền tố lớn nhất bằng 0, tổng của chuỗi tiền tố bằng 0, chỉ số index i bằng 0, sum bắt đầu chạy bằng 0
* Thực hiện vòng lặp: lấy địa chỉ của từng phần tử của mảng thông qua $t3 và lấy giá trị của mảng vào $t4 (lw), cộng giá trị của A[i] ($t4) vào sum ($t1). So sánh sum đang có ($t1) với max sum ($v1). Nếu max sum ($v1) nhỏ hơn sum ($t1) thì cập nhập độ dài chuỗi tiền tố ($v0) +1, max sum mới ($v1) được cập nhập. Nếu không thì thực hiện so sánh xem index i($t0) với n ($a1). Nếu i > n thì thoát vòng lặp, nếu không thực hiện vòng lăp. Max sum và độ dài chuỗi tiền tố cuối cùng chính là giá trị cần tìm

**Bài 2:**

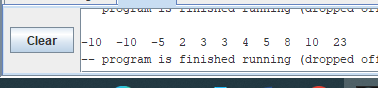






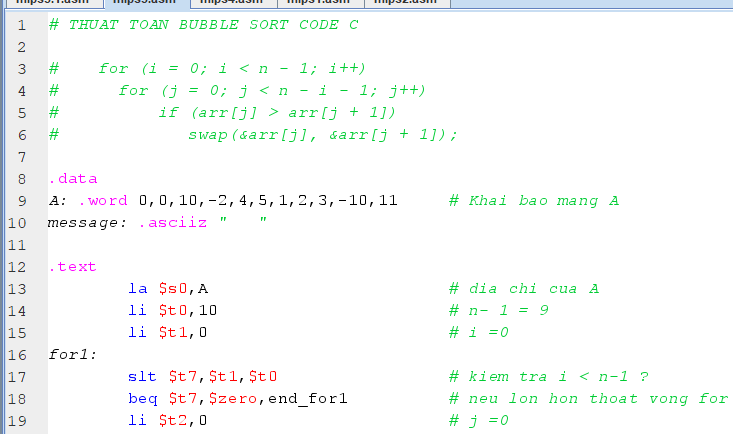
*Thực hiện gõ chương trình với công cụ MARS*

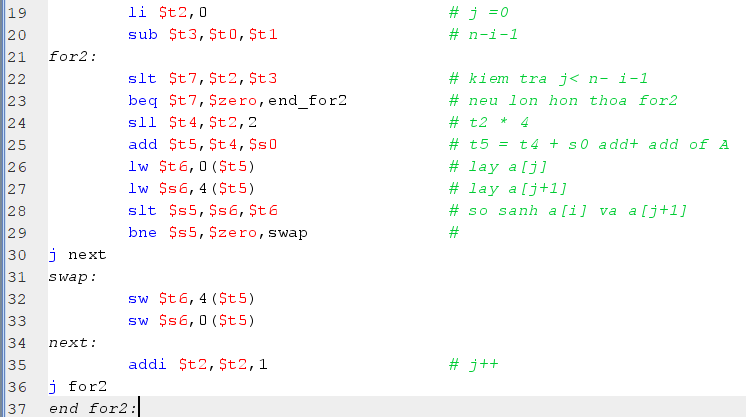
* Kết quả :

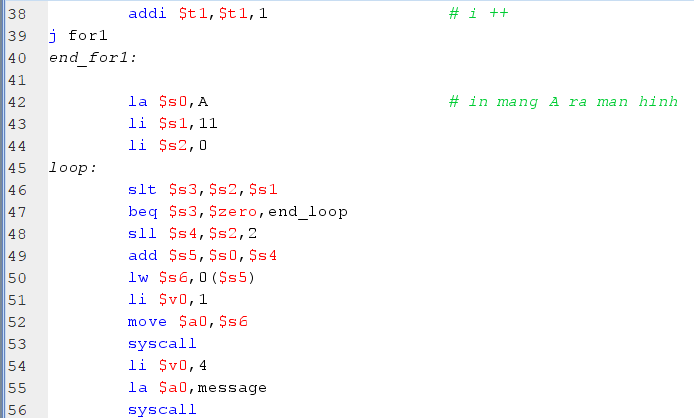


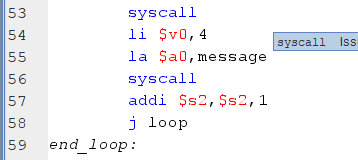
* Giải thích :
  + Khai báo : Mảng A
  + Ở đây lấy địa chỉ phần tử đầu tiên mảng A và địa chỉ cuối mảng A ta lấy thông qua biến Aend bởi vì bộ nhớ được câp phát liên tiếp nên lấy địa chỉ Aend – 4 sẽ ra địa chỉ cuối của A
  + Kiểm tra địa chỉ cuối của A có trùng địa chỉ đầu không. Nếu trùng thì mảng đã được sắp xếp, nếu chưa thì tiếp tục sắp xếp mảng. Thưc hiện vòng lặp để tìm giá trị lớn nhất ở phần chưa được sắp xếp, sau đó đổi chỗ nó về cuối trước phần đã được sắp xếp. Sau mỗi vòng lặp thì địa chỉ cuối của mảng chưa được sắp xếp sẽ được giảm dần -4 . Cứ thực hiện như vậy n-1 lần ta được mảng đã được sắp xếp như kết quả.

**Bài 3:**



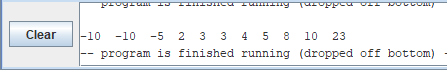






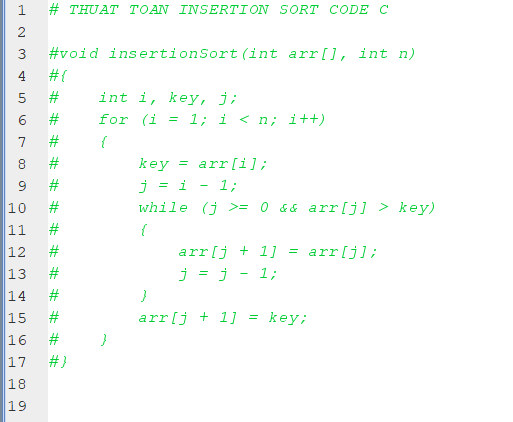
*Thực hiện gõ chương trình với công cụ MARS*

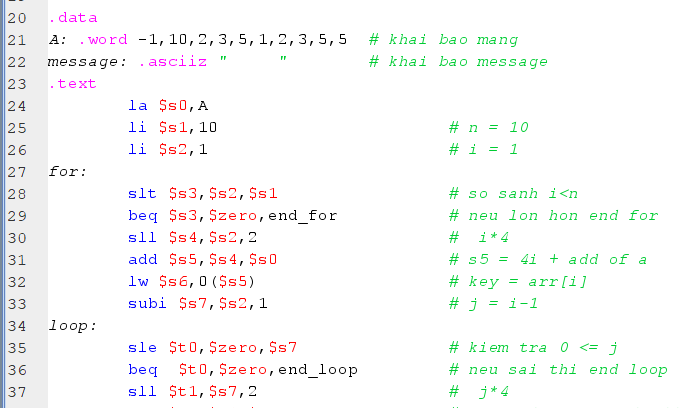
* Kết quả :

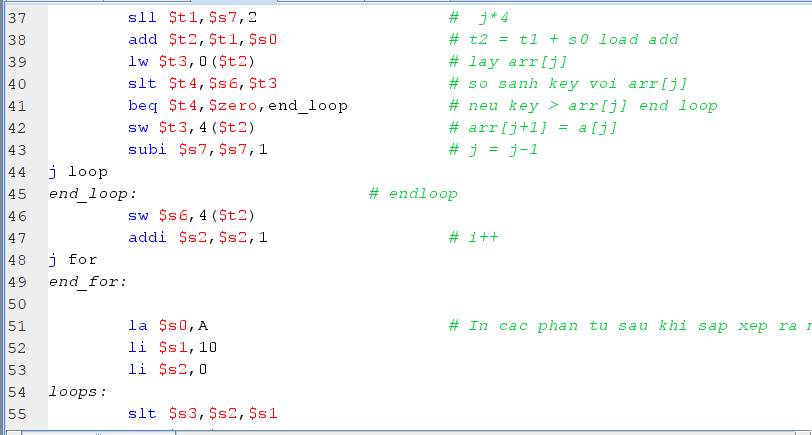


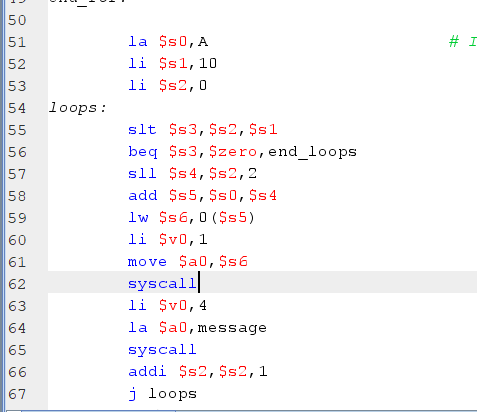
* Giải thích :
  + Khai báo : Mảng A
  + Thực hiện duyệt mảng từ 0 cho đến n-1 (for1). Với mỗi vòng lặp như vậy ta thực hiện vòng lặp for2 cho j chạy từ 0 đến n-1-i. Với mỗi lần lặp thực hiện so sánh a[j] có lơn hơn a[j+1] hay không. Nếu lớn hơn thì thực hiện đổi chỗ hai số đó cho nhau qua lệnh lw và sw.
  + Sau đó thực hiện in mảng ra màn hình (dòng 42-59)
  + Kết quả cho ra 1 mảng đã được sắp xếp theo kiểu Bubble sort

**Bài 4:**



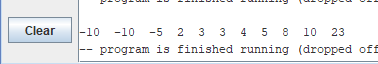






*Thực hành chương trình bằng công cụ MARS*

* Kết quả :



* Giải thích :

Thực hiện vòng lặp for cho i chạy từ 1 đến n. Với mỗi vòng lặp sẽ thực hiện gán key = a[i] thông qua lw. Cho j = i-1 , thực hiện vòng lặp loop, so sánh các điều kiện nếu j < 0 hoặc a[j]<= key thì thoát vòng lặp loop. Nếu không thì thực hiện gán a[j+1] = a[j] bằng lệnh sw. Thực hiện cập nhập j = j – 1. Sau đó thoát vòng lặp loop và cập nhập a[j+1] = key. Thực hiện vòng lặp for cho khi i = n thì dừng

Kết quả cho ra mảng đã sắp xếp theo kiểu insertion sort.