**BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH**

**LAB 6**

Họ và tên: Phạm Vân Anh

MSSV: 20214988

Mã lớp: 139365

***ASSIGNMENT 1:***

1. **Code**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

*Thực hiện gõ chương trình vào công cụ MARS*

1. Giải thích

* Hàm main:
* Gán địa chỉ của A vào $a0
* Gán số phần tử của mảng A vào $a1
* Hàm mspfx: Tìm max sum prefix
* Gán chiều dài của mảng con có tổng lớn nhất vào $v0
* Gán tổng của mảng con có tổng lớn nhất vào $v1
* Gán index i = $t0
* Gán tổng của i phần tử đầu tiên vào $t1
* Hàm loop:
* Dòng 35 🡪 43: Thực hiện truy cập địa chỉ A[i]
* Dòng 38: lw $t4, 0($t3): load giá trị của địa chỉ $t3 vào $t4
* Dòng 40: tính tổng i phần tử đầu tiên, lưu vào thanh ghi $t1
* So sánh với max vừa tìm được:

- Nếu lớn hơn max 🡪 cập nhật max mới (mdfy):

+ Dòng 45: Tăng chiều dài của mảng con vừa tìm được thêm 1

+ Dòng 46: max = tổng hiện tại

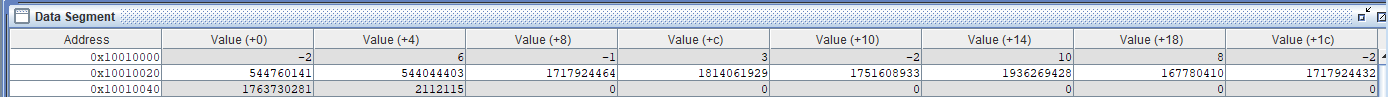
- Nếu nhỏ hơn max (test) 🡪 kiểm tra xem A[i] có phải phần tử cuối cùng trong mảng không. Nếu không thì tiếp tục vòng lặp

* In ra các giá trị: Dòng 55 🡪 69
* Dòng 55 🡪 61: In ra tổng của dãy con có giá trị lớn nhất
* Dòng 63 🡪 69: In ra số giá trị của dãy con

1. Thử với mảng A= { -2, 6, -1, 3, -2, 10, 8, -2}

Graphical user interface, text, application

Description automatically generated



**Kết quả:**

Table

Description automatically generated

***ASSIGNMENT 2:***

1. **Code**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

*Thực hiện gõ chương trình vào công cụ MARS*

1. **Giải thích code**

* Injection Sort:
* Mỗi lần duyệt tìm được phần tử lớn nhất 🡪 cho về cuối mảng
* Gán $s0 trỏ tới phần tử đầu tiên của mảng
* Gán $a1: trỏ tới phần tử cuối cùng của mảng
* Gán $v0: biến con trỏ lưu địa chỉ của phần tử có giá trị lớn nhất từ dãy chưa xếp
* Gán $v1: Giá trị của phần tử lớn nhất chưa được sắp xếp
* Hàm sort: Kiểm tra điều kiện dừng của vòng lặp: Dòng 46, 47
* So sánh $a0 và $a1

- Nếu a0 = a1 🡪 done

- Nếu a0 != a1 🡪 max

* Hàm print\_loop: In ra mảng sau khi sắp xếp
* Hàm after\_max: Đổi chỗ phần tử lớn nhất hiện tại với phần tử cuối cùng của mảng
* Lấy giá trị từ địa chỉ của a1 🡪 t0
* Lưu giá trị của t0 vào v0
* Lưu giá trị của v1 vào a1
* Sau đó, a1 giảm đi 4 để đến phần tử tiếp theo
* 🡪 sort
* Hàm loop:
* Kiểm tra con trỏ có nằm ở cuối không

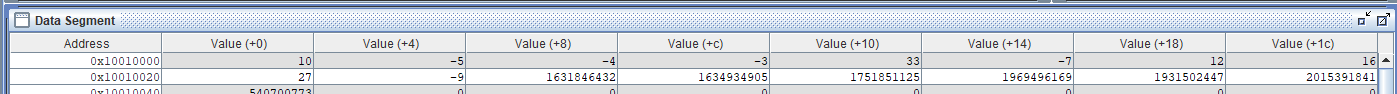
- Có: Vừa hoàn thành 1 vòng lặp 🡪 after\_max

- Không: Tăng con trỏ thêm 4 🡪 sang phần tử tiếp. Kiểm tra xem phần tử đó có lớn hơn max không, nếu không 🡪 loop. Nếu có 🡪 cập nhật max 🡪 loop

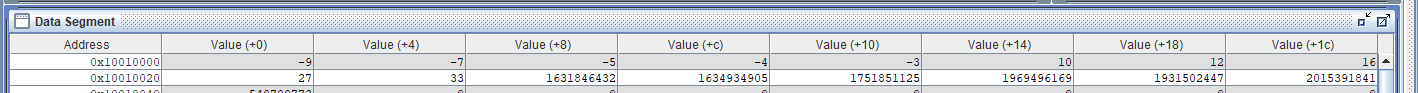
* Hàm after\_sort: Dòng 16 🡪 21
* Gán địa chỉ của A vào $s0
* Gán địa chỉ của Aend (đánh dấu kết thúc mảng) vào $a1
* Lấy a1 = a1 – 4 (a1= address A[n-1])
* Dòng 19 🡪 21: In ra message: “Chuoi sau khi duoc sap xep: “

1. Kết quả:

* Trước khi sắp xếp:



* Sau khi sắp xếp:



* Kết quả:

Text

Description automatically generated

***ASSIGNMENT 3:***

* + - 1. **Code**

**Text

Description automatically generated**

**Text

Description automatically generated**

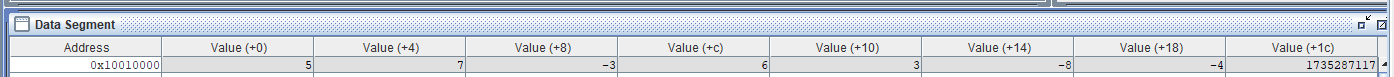
*Thực hiện gõ chương trình vào công cụ MAR*

1. Giải thích code:

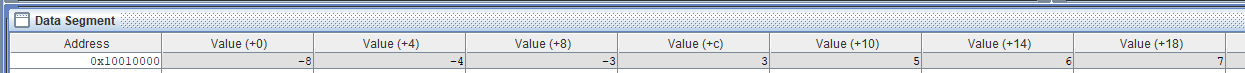
* Hàm sort: Duyệt từ phần tử đầu tiên tới phần tử cuối cùng
* Dòng 25: Duyệt lại từ đầu mảng (sau khi tìm được phần tử lớn nhất đưa về cuối dãy)
* So sánh 2 phần tử liền nhau. Nếu số sau < số hiện tại 🡪 swap
* Hàm swap:
* Gán giá trị của $t1 vào $t5
* Hoàn đổi giá trị của current và next thông qua $t5
* Tăng $v0 (current) trỏ vào các phần tử tiếp theo

1. Kết quả:

* Trước khi duyệt:



* Sau khi duyệt:



* Kết quả:

Graphical user interface, text, application, email

Description automatically generated

***ASSIGNMENT 4:***

**Code**

**Graphical user interface, text

Description automatically generated**

**Text

Description automatically generated**

**Text, application

Description automatically generated**

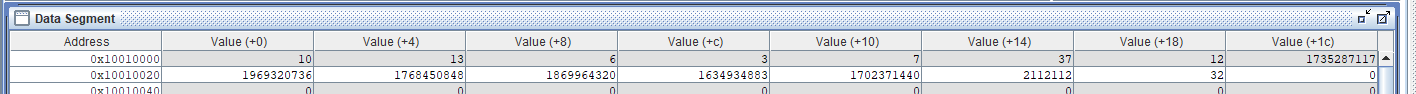
*Thực hiện gõ chương trình vào công cụ MARS*

* + - 1. **Giải thích**
* Gán $a0 mang địa chỉ của A
* Gán $s1 mang địa chỉ cuối mảng
* Hàm sort:
* Duyệt từ phần tử A[1], so sánh với các phần tử trước đó của mảng. Dừng nếu $v0 = $a1

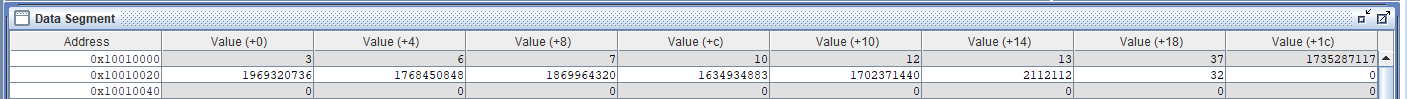
- Nếu số trước lớn hơn 🡪 progress

- Nếu số trước nhỏ hơn 🡪 done

* Hàm progress
* Gán giá trị của phần tử cần sắp xếp lại vào $t5
* Gán $t3 là phần tử bắt đầu đi về sau
  + - 1. **Kết quả:**
* Trước khi xếp:



* Sau khi xếp:



* Kết quả:

Text

Description automatically generated with low confidence