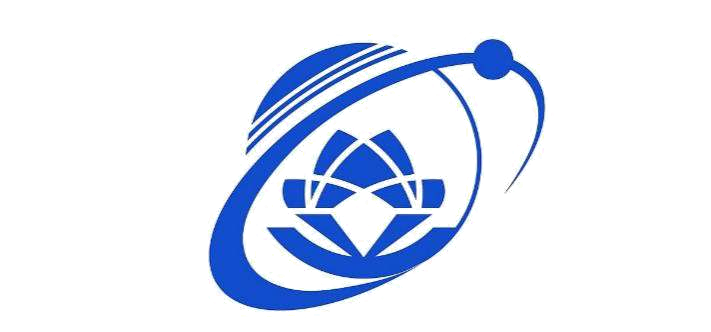
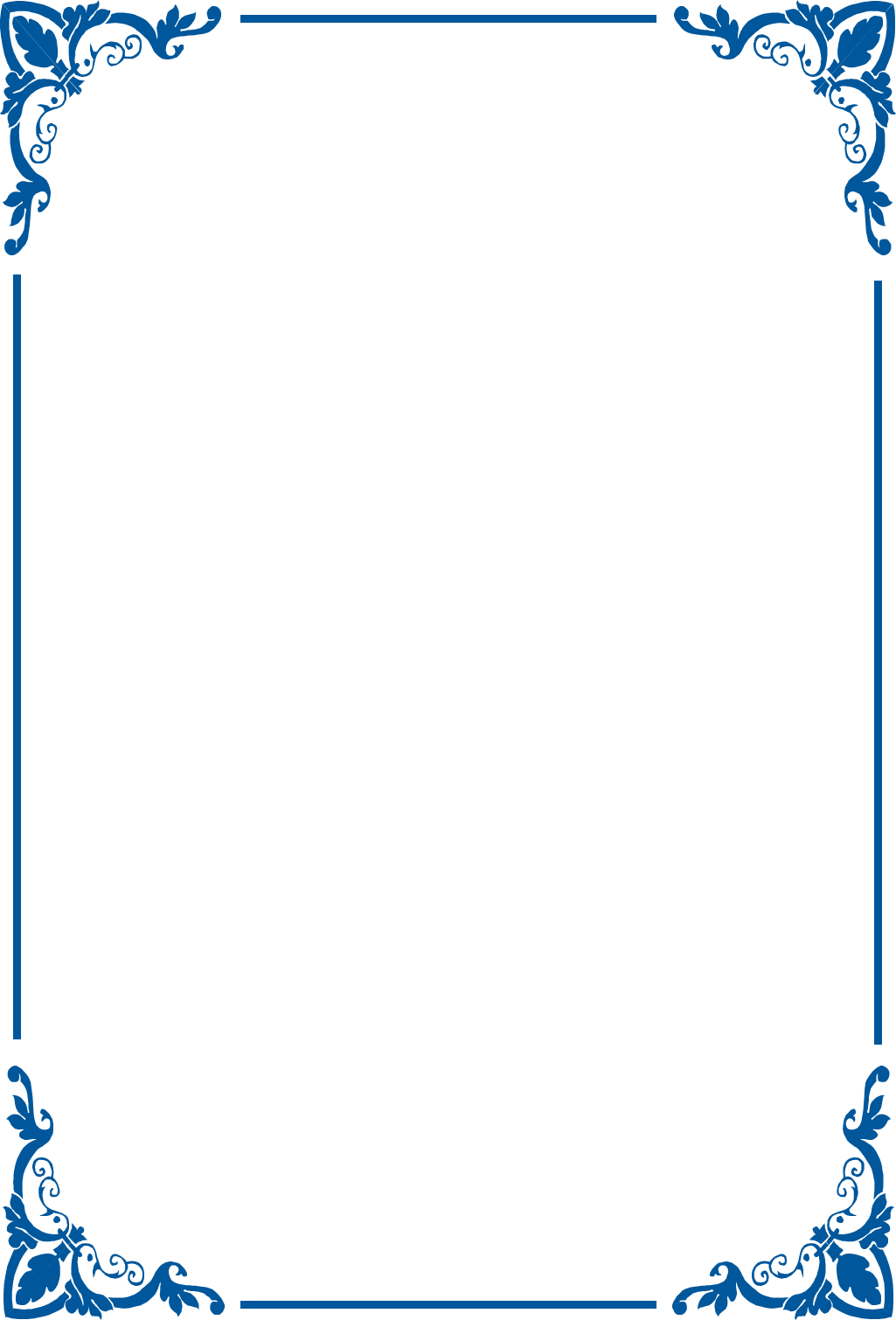
**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**BÁO CÁO BÀI THỰC HÀNH 3:**

**CE119 - THỰC HÀNH KIẾN TRÚC MÁY TÍNH**

**CHỦ ĐỀ**

**CE119-Lab03/IT012-Lab05**

Giảng viên hướng dẫn:ThS. Trần Văn Quang

Sinh viên thực hiện:

Đỗ Minh Hội - 23520550

**1. Thao tác với mảng**

Mảng với n phần tử là một chuỗi n phần tử liên tiếp nhau trong bộ nhớ. Thao tác với

mảng trong MIPS là thao tác trực tiếp với byte/word trong bộ nhớ.

▪ Để cấp phát chuỗi word hoặc byte trong bộ nhớ, có giá trị khởi tao sử dụng

“.word” hoặc “.byte” trong “.data”

▪ Để cấp phát chuỗi byte không có giá trị khởi tạo trước, sử dụng “.space” trong

“.data”

Cho ba mảng với cấp phát dữ liệu trong bộ nhớ như sau:

|  |
| --- |
| .data  array1: .word 5, 6, 7, 8, 1, 2, 3, 9, 10, 4  size1: .word 10  array2: .byte 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16  size2: .word 16  array3: .space 8  size3: .word 8 |

Mảng array1 có 10 word, kích thước được lưu trong size1; Mảng array2 có 16 byte,

kích thước được lưu trong size2; Mảng array3 có 8 byte, kích thước được lưu trong

size3.

Viết code trong phần “.text” thực hiện riêng từng phần việc:

✓ In ra cửa sổ I/O của MARS tất cả các phần tử của mảng array1 và array2

✓ Gán các giá trị cho mảng array3 sao cho

array3[i] = array2[i] + array2[size2 - 1 - i]

✓ Người sử dụng nhập vào mảng thứ mấy và chỉ số phần tử cần lấy trong mảng

đó, chương trình xuất ra phần tử tương ứng.

**2. Thao tác với con trỏ**

Con trỏ là một biến lưu địa chỉ của một biến khác. Thao tác với con trỏ trong MIPS là thao tác trực tiếp với địa chỉ bộ nhớ.

Thực hiện lại các yêu cầu của nội dung 1 với con trỏ.

**3. Bài tập (chỉ sử dụng con trỏ)**

a. Nhập một mảng các số nguyên n phần tử (nhập vào số phần tử và giá trị của

từng phần tử), xuất ra cửa sổ I/O của MARS theo từng yêu cầu sau:

✓ Xuất ra giá trị lớn nhất và nhỏ nhất của mảng

✓ Tổng tất cả các phần tử của mảng

✓ Người sử dụng nhập vào chỉ số của một phần tử nào đó và giá trị của phần tử

đó được in ra cửa sổ

**\* Ý tưởng**

**1**. **Nhập số lượng phần tử của mảng**:

* Chương trình hiển thị yêu cầu người dùng nhập số lượng phần tử n của mảng Array. Nếu n <= 0, chương trình sẽ kết thúc.

**2.** **Nhập giá trị cho từng phần tử của mảng**:

* Chương trình sử dụng một vòng lặp để yêu cầu người dùng nhập giá trị cho từng phần tử của mảng từ Array[0] đến Array[n-1]. Các phần tử này được lưu vào mảng array trong bộ nhớ.

**3.** **Tìm giá trị lớn nhất, nhỏ nhất và tính tổng các phần tử của mảng**:

* Sau khi nhập xong các phần tử, chương trình duyệt qua mảng để tìm giá trị lớn nhất (max), giá trị nhỏ nhất (min), và tính tổng (sum) của các phần tử:
  + Biến t4 lưu giá trị lớn nhất ban đầu là Array[0].
  + Biến t5 lưu giá trị nhỏ nhất ban đầu là Array[0].
  + Biến t6 lưu tổng của các phần tử.
  + Vòng lặp duyệt từ Array[1] đến Array[n-1], so sánh từng phần tử với giá trị lớn nhất và nhỏ nhất hiện tại, và cập nhật nếu cần. Đồng thời, tổng được cập nhật liên tục.

**4.** **Hiển thị kết quả**:

* Sau khi tính toán, chương trình hiển thị giá trị lớn nhất, giá trị nhỏ nhất, và tổng các phần tử của mảng ra màn hình.

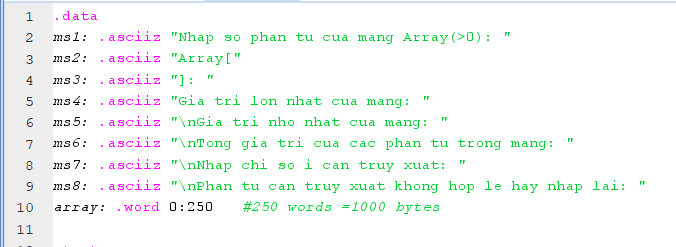
**5.** **Truy xuất phần tử tại chỉ số i do người dùng nhập**:

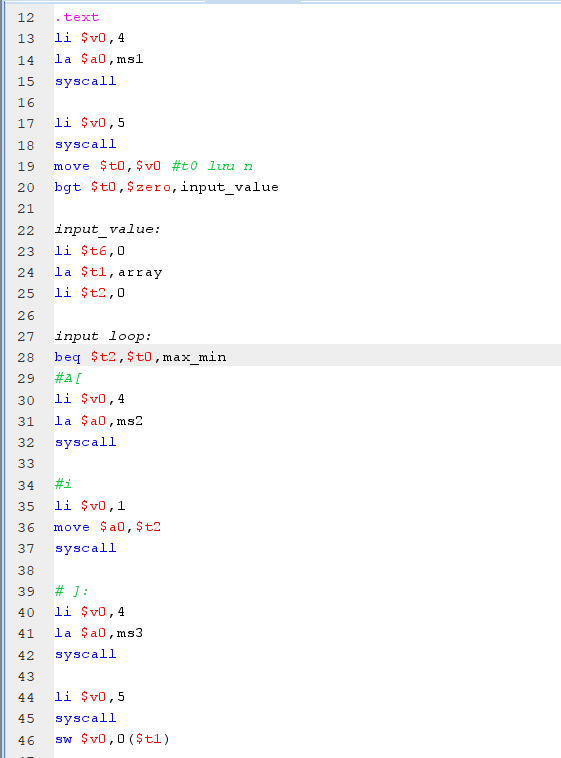
* Chương trình yêu cầu người dùng nhập chỉ số i muốn truy xuất trong mảng Array. Nếu chỉ số i không hợp lệ (ngoài phạm vi [0, n-1]), chương trình sẽ yêu cầu nhập lại và hiển thị thông báo lỗi.
* Khi chỉ số i hợp lệ, chương trình lấy và in ra giá trị của phần tử tại Array[i].

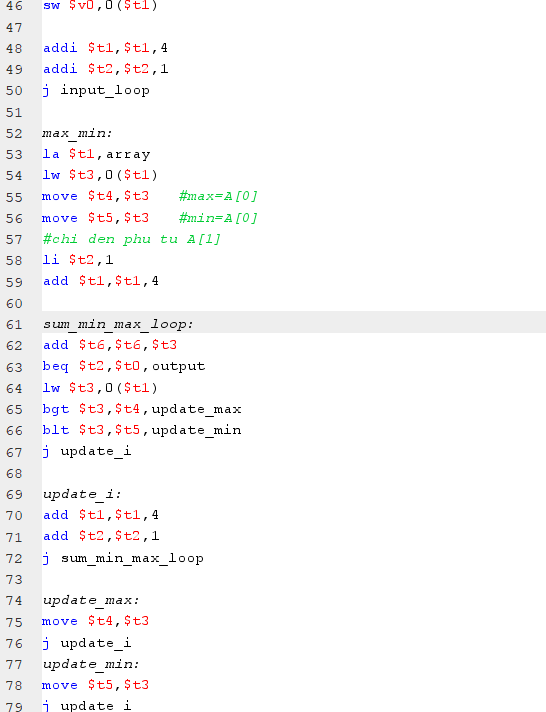
**6.** **Kết thúc chương trình**:

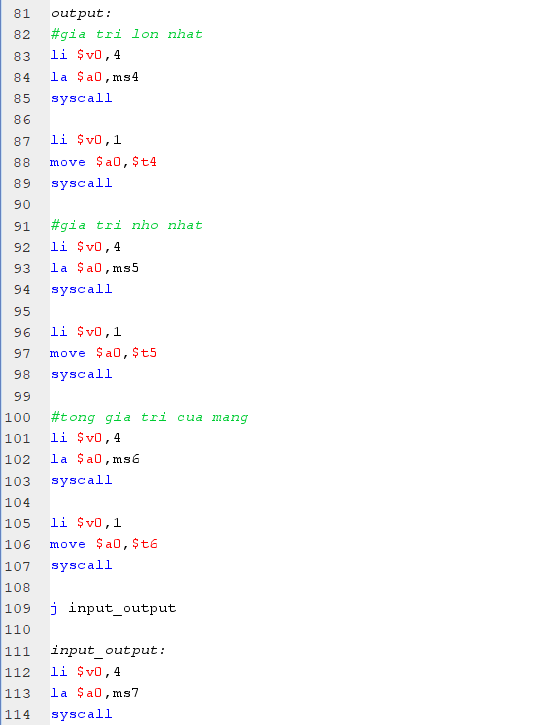
* Sau khi hoàn thành các yêu cầu, chương trình thoát.

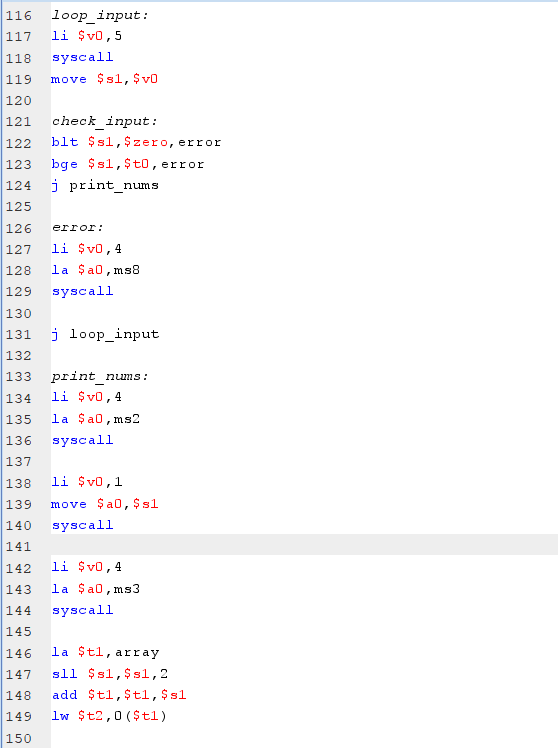
**-Thực hiện viết chương trình**

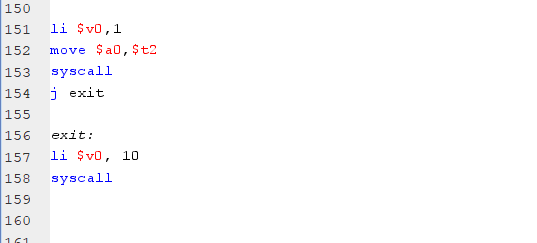




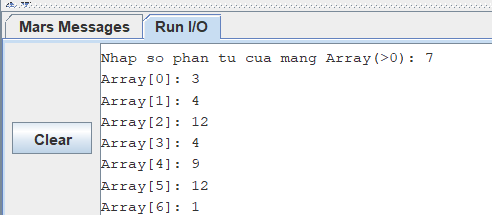


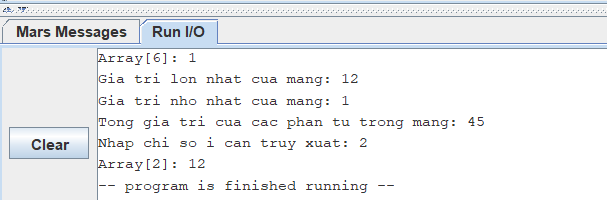




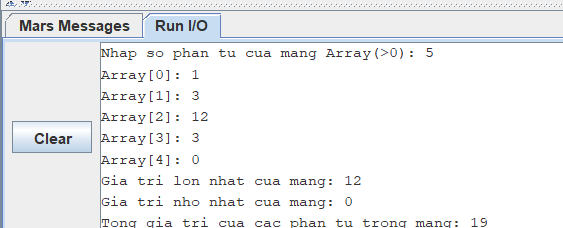


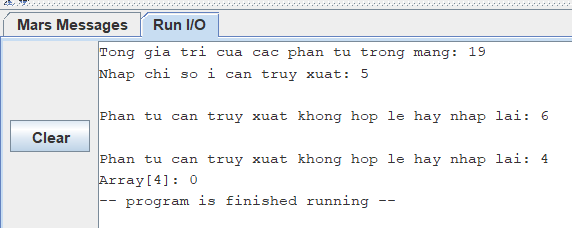
**-Thực hiện chạy chương trình**

****

****

+trường hợp nhập i>= số phần tử của mảng thì yêu cầu người dùng nhập lại

****

****

b. Nhập một mảng các số nguyên n phần tử (nhập vào số phần tử và giá trị của

từng phần tử). Mảng này gọi là A.

Chuyển dòng lệnh C dưới đây sang mã assembly của MIPS. Với các biến nguyên

i, j được gán lần lượt vào thanh ghi $s0, $s1; và địa chỉ nền của mảng số nguyên A

được lưu trong thanh ghi $s3

|  |
| --- |
| if (i<j) A[i]= i;  else A[i] = j; |

**\* Ý tưởng:**

1 **Nhập số lượng phần tử của mảng**:

* Đầu tiên, chương trình hiển thị lời nhắc yêu cầu người dùng nhập số lượng phần tử n cho mảng A. Số lượng này phải lớn hơn 0. Nếu người dùng nhập n <= 0, chương trình sẽ thoát.

2 **Nhập các phần tử của mảng A**:

* Sau khi nhận được n, chương trình sử dụng một vòng lặp để yêu cầu người dùng nhập từng phần tử của mảng A, từ A[0] đến A[n-1]. Các phần tử này được lưu vào mảng array trong bộ nhớ.

3 **Nhập chỉ số i và j**:

* Chương trình yêu cầu người dùng nhập hai chỉ số i và j. Mỗi chỉ số được kiểm tra để đảm bảo rằng nó nằm trong khoảng [0, n-1]. Nếu chỉ số nào không hợp lệ, chương trình sẽ yêu cầu nhập lại chỉ số đó và hiển thị thông báo lỗi tương ứng.

4 **So sánh và gán giá trị**:

* Sau khi có giá trị hợp lệ cho i và j, chương trình thực hiện so sánh i và j:
  + Nếu i < j, chương trình sẽ gán A[i] = i.
  + Nếu i >= j, chương trình sẽ gán A[i] = j.
* Việc này được thực hiện bằng cách tính địa chỉ của A[i] trong bộ nhớ và lưu giá trị i hoặc j vào vị trí đó.

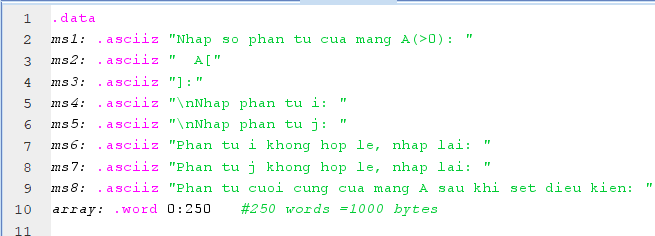
5 **Hiển thị mảng sau khi gán**:

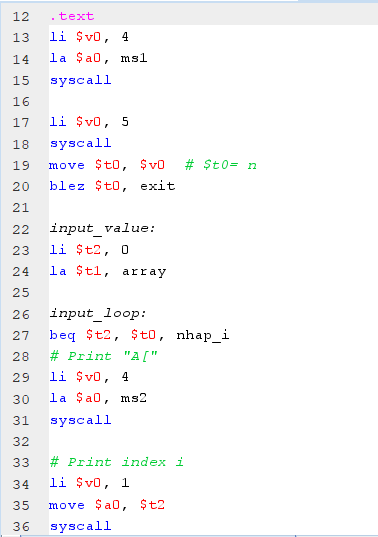
* Cuối cùng, chương trình hiển thị toàn bộ mảng A với các giá trị đã được gán lại. Chương trình duyệt từ A[0] đến A[n-1], in từng phần tử ra màn hình kèm theo chỉ số của nó.

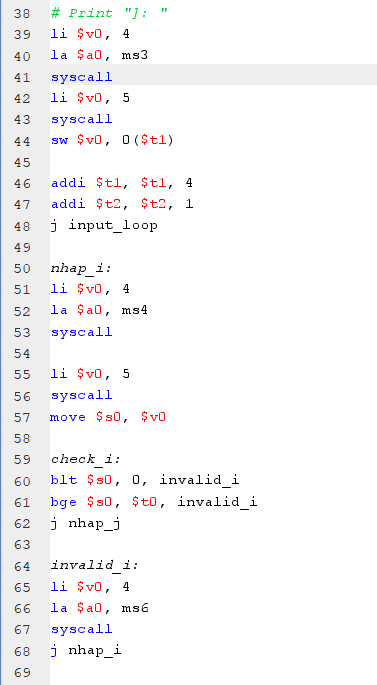
6 **Kết thúc chương trình**:

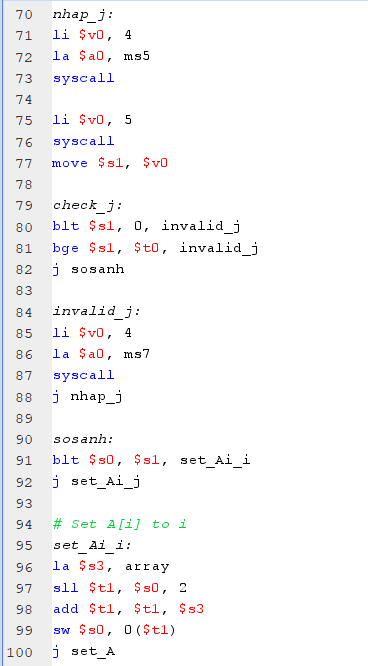
* Sau khi hoàn tất các bước trên, chương trình sẽ thoát.

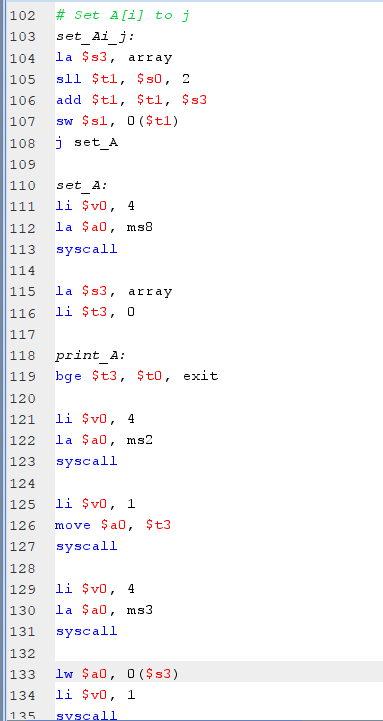
**-Thực hiện viết chương trình**

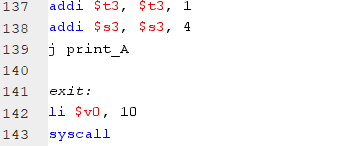






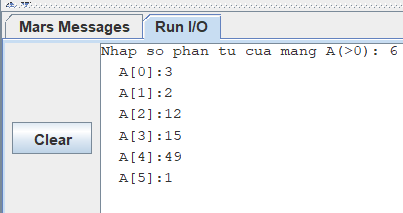


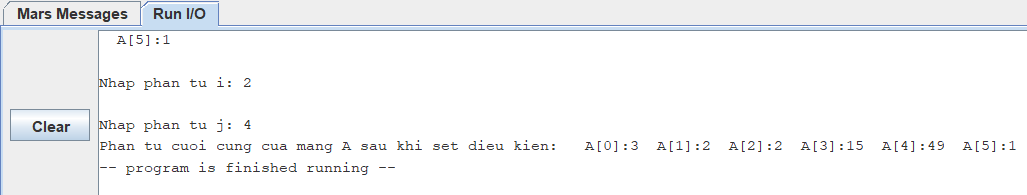




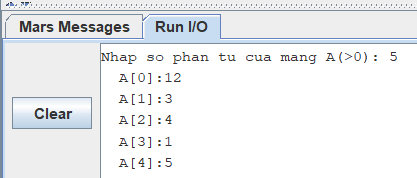
-Thực hiện chạy chương trình

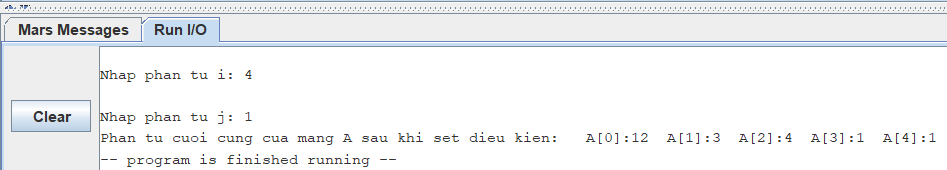
+Vì i(2)<j(4) nên A[2]=2 ta thấy output đã thay đổi giá trị A[2]=12 thành A[2]=2





+Vì i(4)>j(1) nên A[4]=1 ta thấy output đã thay đổi giá trị A[4]=5 thành A[4]=1





+Trường hợp nhập i và j lỗi cũng sẽ được nhập lại

