Bài thực hành giữa kỳ

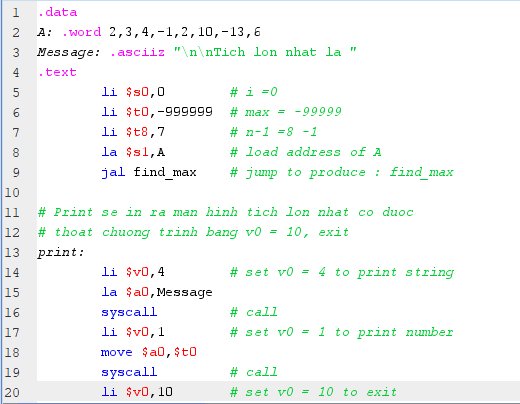
# Lớp : 139365 – Học phần: Thực hành Kiến trúc máy tính

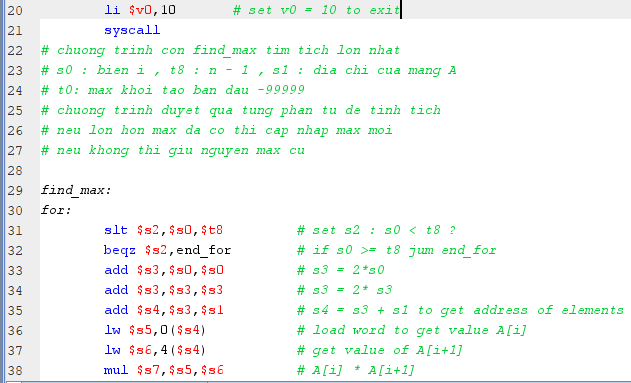
# Họ và tên : Nguyễn Thị Thùy Dung MSSV : 20215009

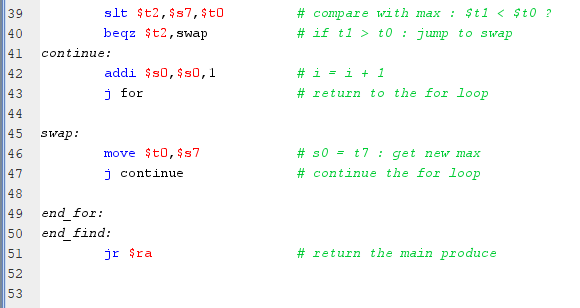
# Họ và tên : Nguyễn Văn Dũng MSSV :20215013

# Project 6:

# Given an array of .word elements and the number of elements, write a procedure to find the pair of adjacent elements that has the largest product and return that product. Example: For inputArray = [3, 6, -2, -5, 7, 3], the output should be the product of 7 and 3 (21)





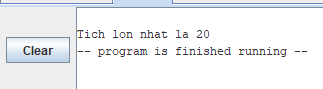


***Thực hiện chạy thử chương trình với MIPS***

* Giả thiết :



* Kết quả :



* Giải thích :

1. Khai báo và khởi tạo các biến:

Khởi tạo biến đếm i ($s0) bằng 0.Khởi tạo biến max($t0) với giá trị ban đầu là

- 999999. Khởi tạo biến n-1 ($t8) bằng 7 . Gán địa chỉ của mảng A cho thanh ghi s1.

2. Gọi hàm find\_max để tìm tích lớn nhất:

- jal find\_max: Gọi hàm find\_max.

3. In ra màn hình tích lớn nhất:

- print:

- li $v0, 4: Thiết lập giá trị 4 cho thanh ghi $v0 để in chuỗi.

- la $a0, Message: Đặt địa chỉ của chuỗi Message vào thanh ghi $a0.

- syscall: Gọi hệ thống để in chuỗi.

- li $v0, 1: Thiết lập giá trị 1 cho thanh ghi $v0 để in số.

- move $a0, $t0: Gán giá trị của max vào thanh ghi $a0.

- syscall: Gọi hệ thống để in số.

- li $v0, 10: Thiết lập giá trị 10 cho thanh ghi $v0 để thoát chương trình.

- syscall : Gọi hệ thống để thoát.

4. Hàm find\_max:

- Vòng for:

- slt $s2, $s0, $t8: So sánh i($s0) và n-1($t8). Nếu i < n-1, kết quả sẽ được gán vào thanh ghi $s2.

- beqz $s2, end\_for: Nếu i >= n-1, nhảy tới nhãn end\_for để kết thúc vòng lặp.

- add $s3, $s0, $s0: $s3 = $s0 + $s0

- add $s3, $s3, $s3: $s3 = $s3 + $s3

- add $s4, $s3, $s1: $s4 = $s3 + $s1

- lw $s5, 0($s4): Load giá trị của A[i] vào thanh ghi $s5.

- lw $s6, 4($s4): Load giá trị của A[i+1] vào thanh ghi $s6.

- mul $s7, $s5, $s6: Nhân $s5 và $s6 và lưu kết quả vào thanh ghi $s7.

- slt $t2, $s7, $t0: So sánh kết quả tích với max. Nếu kết quả nhỏ hơn max, kết quả sẽ được gán vào thanh ghi $t2.

- beqz $t2, swap: Nếu kết quả tích lớn hơn max, nhảy tới nhãn swap để cập nhật max.

- addi $s0, $s0, 1: Tăng giá trị của i lên 1.

- j for: Quay lại vòng lặp for.

- move $t0, $t1: Gán giá trị của $t1 vào max.

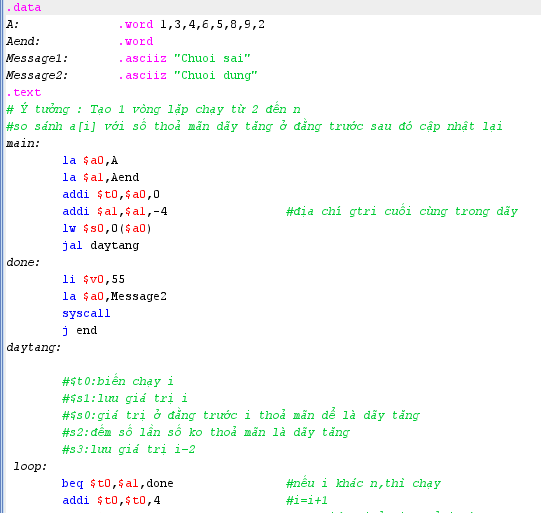
- j continue: Tiếp tục vòng lặp for.

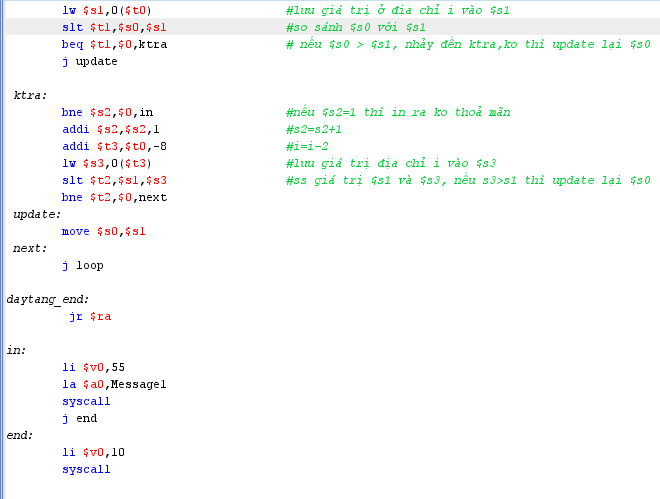
- jr $ra: Lệnh trả về từ hàm và quay lại chương trình chính.

* Chương trình duyệt qua từng phần tử của mảng, tính tích 2 phần tử liên tiếp rồi so sánh với max nếu lớn hơn max thì cập nhập max nếu không thì giữ nguyên max

Project 16 :

Given a sequence of integers as an array, determine whether it is possible to obtain a strictly increasing sequence by removing no more than one element from the array.  
Note: sequence a0, a1, ..., an is considered to be strictly increasing if a0 < a1 < ... < an. Sequences containing only one element are also considered to be strictly increasing.  
Example:  
- For sequence = [1, 3, 2, 1], the output should be almostIncreasingSequence(sequence) = false. There is no one element in this array that can be removed in order to get a strictly increasing sequence.  
- For sequence = [1, 3, 2], the output should be almostIncreasingSequence(sequence) = true. You can remove 3 from the array to get the strictly increasing sequence [1, 2]. Alternately, you can remove 2 to get the strictly increasing sequence [1, 3].



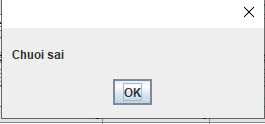


***Thực hiện chạy thử chương trình với MIPS***

* Giả thiết :



* Kết quả :



* Giải thích :

1. Khai báo và khởi tạo các biến:

Lưu array vào $a0, địa chỉ giá trị cuối của array vào $a1,giá trị phần tử đầu của array vào $s0 và gọi hàm daytang

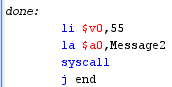
2. Gọi hàm daytang để kiểm tra xem hàm ấy có phải là dãy tănng khi bỏ 1 phần từ không

- jal find\_max: Gọi hàm find\_max.

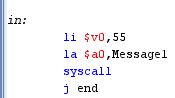
3. In ra màn hình:

Có 2 trường hợp xảy ra:

-Hàm ấy là dãy tăng khi bỏ 1 phần tử:



-Hàm ấy ko là dãy tăng khi bỏ 1 phần tử:



4. Hàm daytang:

Ý tưởng: Ta sẽ có 1 vòng lặp chạy từ 2 đến n-1 , lưu giá trị i vào $s1,so sánh giá trị i với $s0 với $s0 là giá trị ở cuối dãy tăng từ 1 đến i-1. Nếu

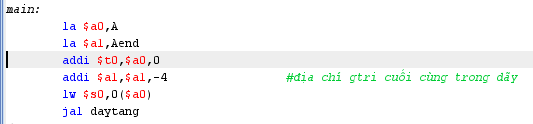
-$s0<$s1:Cập nhật lại $s0= $s1 và quay lại vòng lặp

-$s0>$s1:Gọi $t2 là số lần $s0>$s1.Nếu $t2=1 thì in ra “Chuoi sai”,ngược lại $t2=$t2+1.Gọi giá trị ở i-1 là $s3 Có 2 trường hợp xảy ra

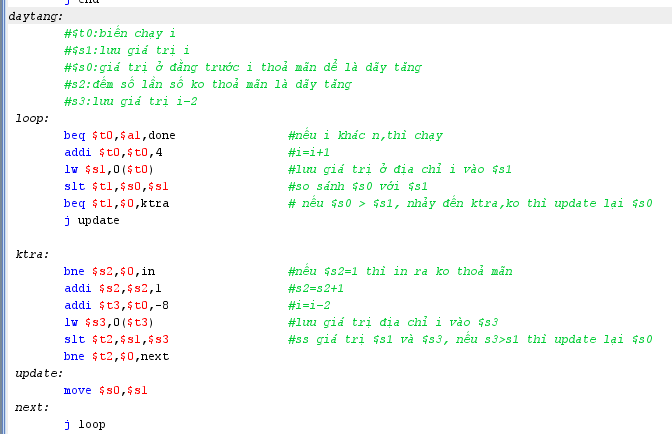
+)$s3<$s1:không phải cập nhật $s0, quay lại vòng lặp

+)$s3>$s1:cập nhập lại $s0=$s1,quay lại vòng lặp

Giải thích:



Gán giá trị của array vào $a0, và $a1 là địa chỉ giá trị cuối cùng trong dãy. Gọi $s0 là giá trị ở cuối dãy tăng thoả mãn từ 1 đến i-1.$s0 là giá trị $a0. Gán địa chỉ phần tử đầu vào $t0.Sau đó nhảy đến hàm daytang.



-Tạo 1 vòng lặp chạy từ 1 đến n-1 với biến chạy là $t0, nếu $t0 là địa chỉ ở phần tử cuối array, ngừng vòng lặp.Ngược lại, cộng thêm 4 vào $t0 và lấy giá trị từ địa chỉ của $t0 gán vào $s3.

Nếu $s0< $s1 thì nhảy đến update để cập nhập lại $s0=$s1 và quay lại vòng lặp,ngược lại thì nhảy đến ktra để xử lý

-Gọi $s2 biến đếm số lần $s0>$s1. Nêu $s2=1 thì nhảy đến in để in ra “Chuỗi sai”.Hoặc là tăng $s2 lên 1. Gọi $s3 là phần tử thứ i-2 trong dãy tăng thoả mãn từ 1 đến i-1. Nếu

+)$s3<$s1:không phải cập nhật $s0, quay lại vòng lặp

+)$s3>$s1:cập nhập lại $s0=$s1,quay lại vòng lặp

Nếu có dãy tăng thoả mãn thì quay trở lại địa chỉ câu lệnh sau câu lệnh jal daytang. Và in ra “Chuoi dung”.