**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**Architecture Computer - Assembly**

### **Đồ án 2:**

### **Hợp ngữ Mips – xử lí mảng số nguyên**

By

Vũ Hoàng Dương - 1612127

Lê Quốc Dũng – 1612126

Nguyễn Hữu Điền – 1612104

Nguyễn Hoàng Giang – 1612151

**Teacher’s Name**

Lê Viết Long

**Class**: KTMTHN-2016/4

# Ý TƯỞNG, THUẬT TOÁN VÀ CÀI ĐẶT

### **1.NHẬP MẢNG**

a, Đánh giá mức độ hoàn thành

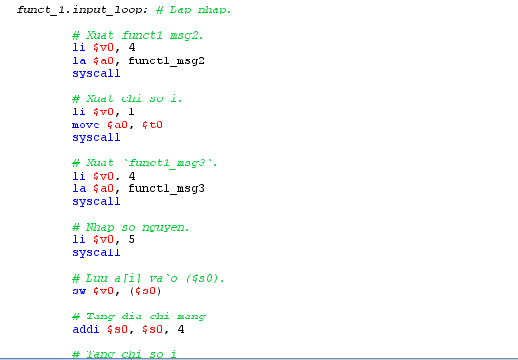
Hoàn thành: 100%

b, Ý tưởng

-Lưu thứ tự các phần tử vào mảng arr\_data.

-Tăng địa chỉ thanh ghi lên +4 sau mỗi lần thêm phần tử

c, Cài đặt



### **2. XUẤT MẢNG**

a, Đánh giá mức độ hoàn thành

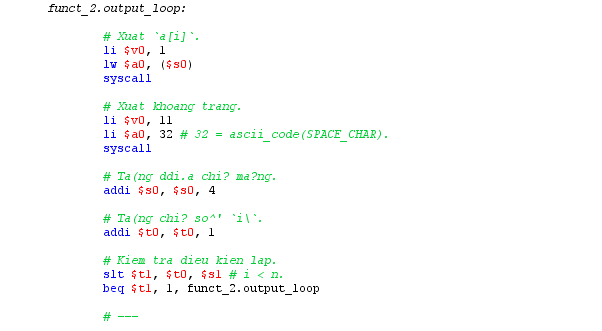
Hoàn thành: 100%

b, Ý tưởng

-Load địa chỉ của phần tử đầu tiên của mảng vào thanh ghi.

-Sau mỗi lần lặp xuất, tăng địa chỉ thanh ghi lên +4

c, Cài đặt



### **3. LIỆT KÊ CÁC SỐ NGUYÊN TỐ TRONG MẢNG**

a. Đánh giá mức độ hoàn thành  
Hoàn thành: 100%

b, Ý tưởng

Tách 2 hàm xử lí:

-Hàm 1: Xét phần tử truyền vào, kiểm tra có phải là số nguyên tố hay không

-Hàm 2: Liệt kê tất cả số nguyên tố có trong mảng.

c, Thuật toán:

Kiểm tra số nguyên tố?

Số nguyên tố là số có 2 ước là 1 và chính nó (ngoại trừ 1 và 2)

-Khởi tạo biến đếm i=1

-Chia phần tử cho i, lấy dư. Kiểm tra dư. Bằng 0? Tăng biến đếm thanh ghi lên 1, tăng i

Khác 0? Tiếp tục tăng i

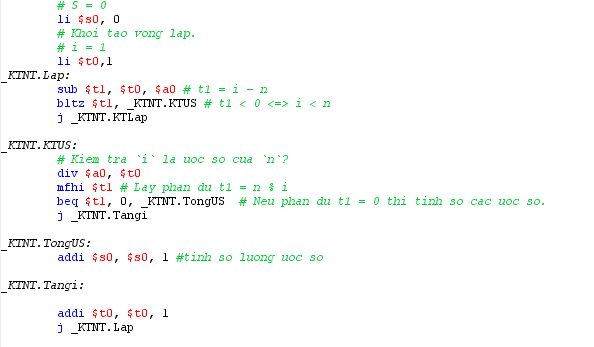
-Lặp đến khi nào i<=n (i phải nhỏ hơn n, ước n sẽ không được tính)

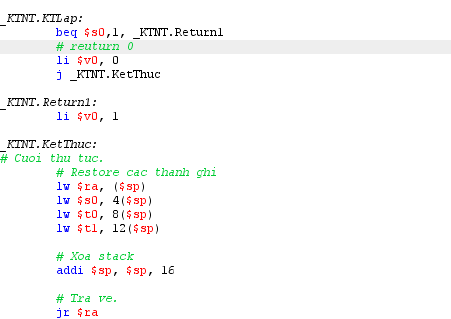
-Nếu số ước trong thanh ghi đếm bằng 1. Trả về 1 (là số nguyên tố)

-Ngược lại, trả về 0 (không là số nguyên tố).

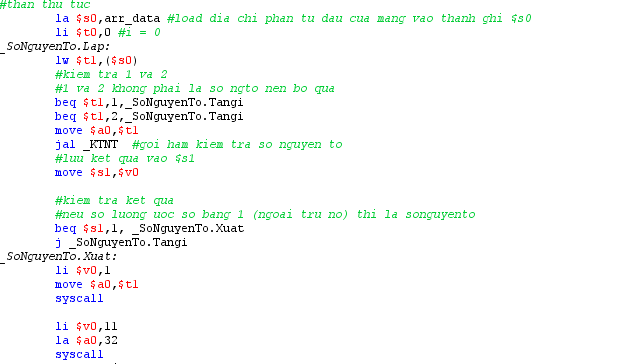
d, Cài đặt

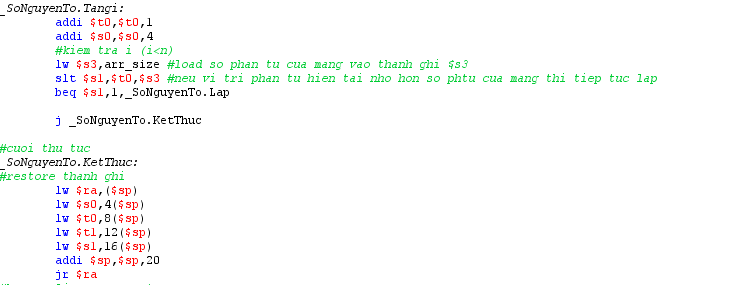
Hàm kiểm tra số nguyên tố:





Hàm liệt kê số nguyên tố:





### **4. LIỆT KÊ CÁC SỐ HOÀN THIỆN TRONG MẢNG**

a, Đánh giá mức độ hoàn thành

Hoàn thành: 100%

b, Ý tưởng

Tách 2 hàm xử lí:

-Hàm 1: Xét phần tử truyền vào, kiểm tra có phải là số hoàn thiện hay không

-Hàm 2: Liệt kê tất cả số hoàn thiện có trong mảng.

c, Thuật toán

Kiểm tra số hoàn thiện?

Số hoàn thiện là số có tổng các ước (ngoại trừ chính nó) bằng chính nó.

-Khởi tạo biến đếm i=1

-Chia phần tử cho i, lấy dư. Kiểm tra dư. Bằng 0? Cộng i vào biến đếm thanh ghi, tăng i

Khác 0? Tiếp tục tăng i

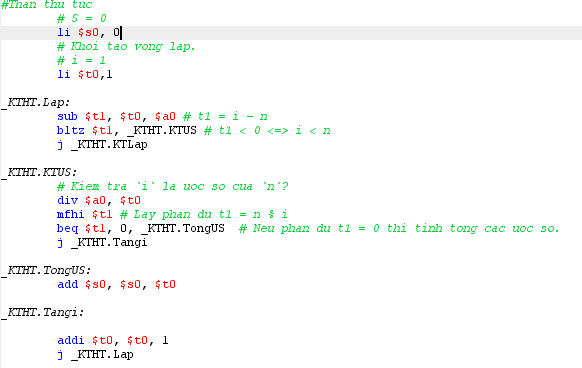
-Lặp đến khi nào i<=n (i phải nhỏ hơn n, ước n sẽ không được tính)

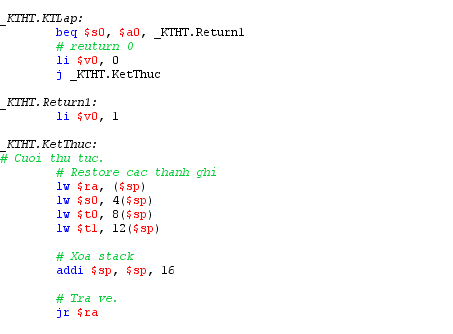
-Nếu tổng ước trong thanh ghi đếm bằng n. Trả về 1 (là số hoàn thiện)

-Ngược lại, trả về 0 (không là số hoàn thiện).

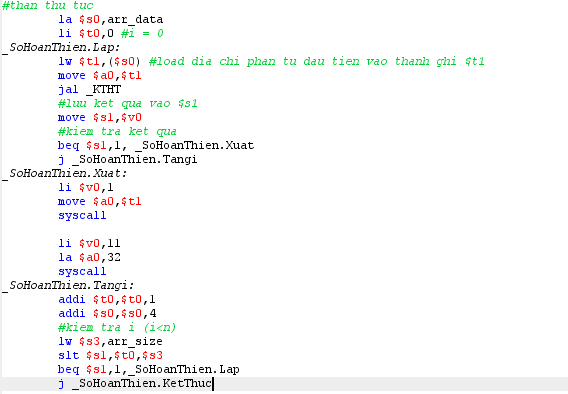
d, Cài đặt

Hàm kiểm tra số hoàn thiện





Hàm liệt kê các số hoàn thiện trong mảng:



### **5. TÍNH TỔNG CÁC SỐ CHÍNH PHƯƠNG TRONG MẢNG**

a, Đánh giá mức độ hoàn thành

Hoàn thành: 100%

b, Ý tưởng

Tách 2 hàm xử lí:

-Hàm 1: Xét phần tử truyền vào, kiểm tra có phải là số chính phương hay không

-Hàm 2: Tính tổng tất cả số chính phương có trong mảng.

c, Thuật toán

Kiểm tra số chính phương?

Số chính phương là bình phương của 1 số tự nhiên.

-Khởi tạo biến đếm i=1

-Tính i^2. Bằng n? Trả về 1 (là số chính phương)

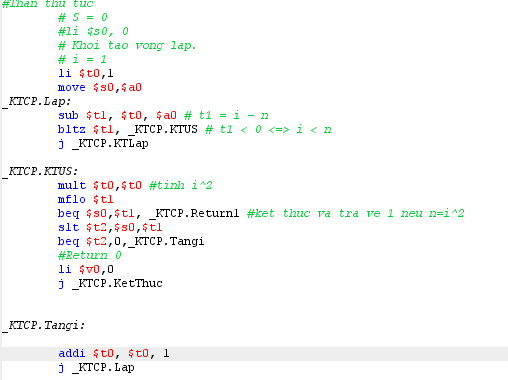
Khác n? Tiếp tục tăng i

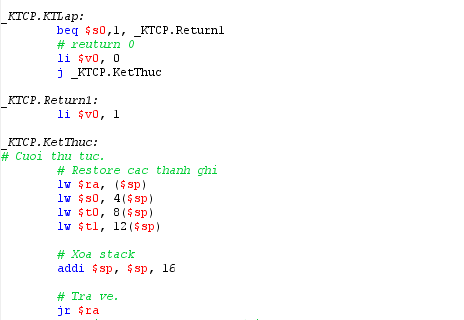
-Lặp đến khi nào i<=n (i phải nhỏ hơn n, ước n sẽ không được tính)

-Trả về 0 (không là số chính phương).

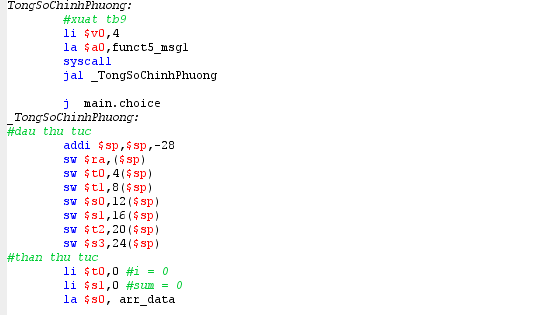
d, Cài đặt

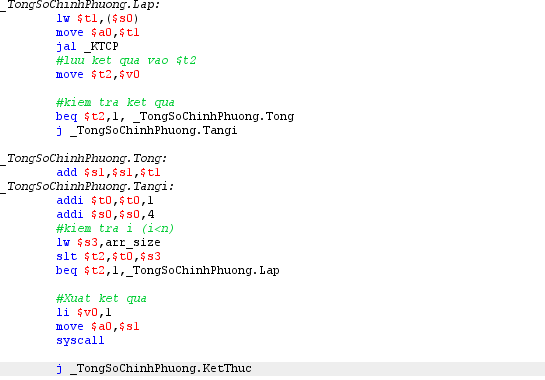
Hàm kiểm tra số chính phương





Hàm tính tổng số chính phương:





### **6. Tính trung bình cộng các số đối xứng**

a. Đánh giá mức độ hoàn thành  
Hoàn thành: 100%

b. Ý tưởng

- Tách thành 2 hàm để xử lý  
+ Hàm 1: Kiểm tra một số nguyên truyền vào là số đối xứng hay không

+ Hàm 2: Tính trung bình cộng của các số đối xứng có trong mảng nhập vào

c. Thuật toán  
- Hàn kiểm tra một số nguyên là số đối xứng:

+ Đảo ngược số nguyên nhập vào bằng cách áp dụng các phép chia lấy dư và chia lấy nguyên  
vd: a= 123 : 123%10=3; b=3; a=123/10=12; b=3\*10+12%10=32; a=12/10=1; b=32\*10+a%10=321  
+ So sanh số sau khi đảo và trước khi đảo nếu bằng nhau là số đối xứng, ngược lại không phải số đối xứng/

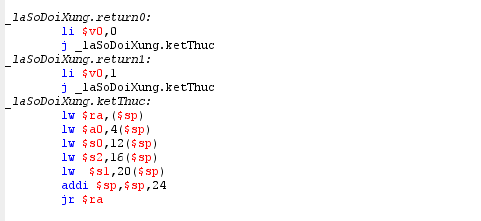
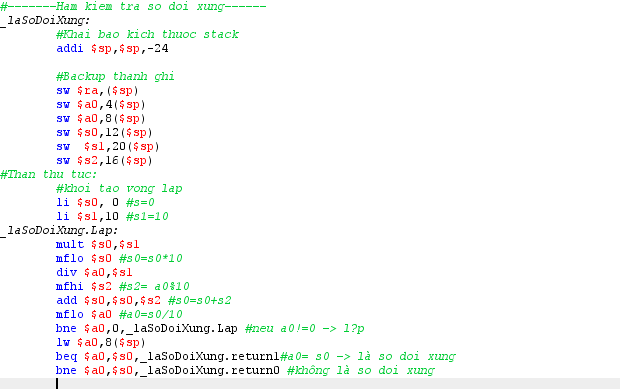
- Hàm tinh trung bình cộng các số đối xứng

+ Duyệt toàn mảng, kiểm tra số nào là số đối xứng, cộng dồn rồi chia cho số lượng số đối xứng có trong mảng.

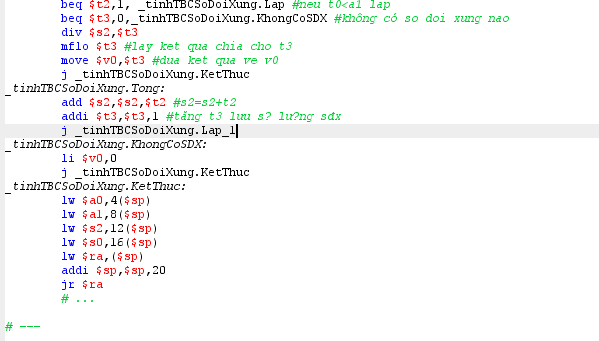
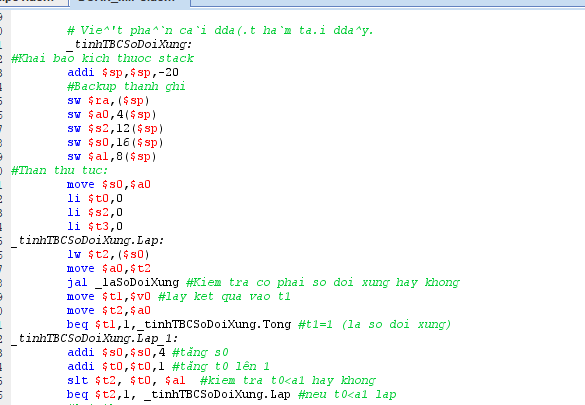
+ Trường hợp không có số đối xứng, trả về 0.s

d. Cài đặt

- Hàm kiểm tra một số là số đối xứng hay không:



- Hàm tinh trung bình cộng các số đối xứng:



### **7. Hàm tìm giá trị lớn nhất trong mảng**

a. Đánh giá mức độ hoàn thành

- Mức độ hoàn thành 100%

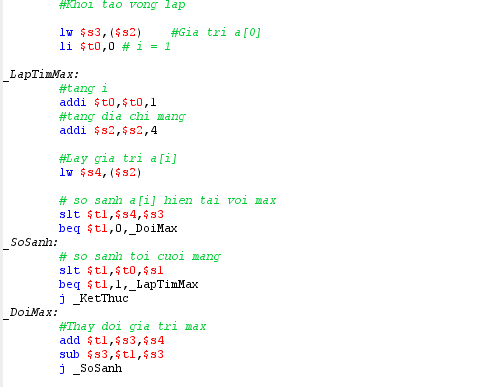
b. Ý tưởng và thuật toán

- Đầu tiên ta sẽ tạo một giá trị Max bằng giá trị tại phần tử đầu tiên của mảng

- Sau đó, ta sẽ duyệt từ phần tử thứ hai của mảng về sau đến hết mảng. Với mỗi lần duyệt như vậy ta sẽ xét xem giá trị tại phần tử đó có lớn hơn giá trị Max hay không, nếu có thì ta sẽ gán giá trị Max bằng giá trị của phần tử đó.

- Sau khi duyệt xong thì ta sẽ tìm được giá trị lớn nhất và trả về .

c. Cài đặt



### **8. Hàm sắp xếp tăng dần theo thuật toán Selection Sort**

a. Đánh giá mức độ hoàn thành

- Mức độ hoàn thành 100%

b. Ý tưởng và thuật toán

- Chúng ta sẽ có hai vòng lặp:

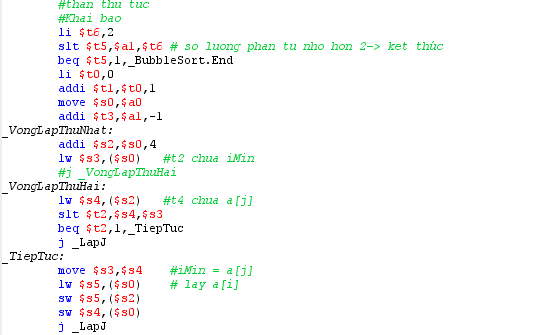
+ Vòng lặp thứ nhất duyệt từ đầu đến phần tử kề cuối mảng. Với mỗi lần duyệt ta sẽ tạo giá trị iMin bằng giá trị phần tử tại đó( phần tử tại đó gọi là a[i])

+ Vòng lặp thứ hai duyệt từ vị trí của phần tử đang xét ở vòng lặp thứ nhất về cuối mảng. Với mỗi lần duyệt ta sẽ tính giá trị phần tử tại đó gọi là a[j] Sau đó ta so sánh giá trị này với iMin.

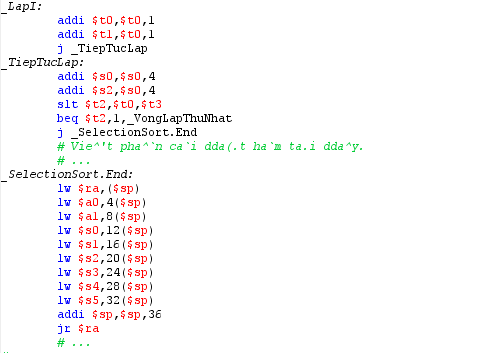
* Nếu a[j]<iMin thì gán iMin = a[j] rồi hoán đổi giá trị a[i] và a[j] với nhau

c. Cài đặt

Vòng lặp thứ nhất:



Vòng lặp thứ hai:



### **9. Hàm sắp xếp giảm dần theo thuật toán Bubble Sort**

a. Đánh giá mức độ hoàn thành

- Mức độ hoàn thành 100%

b. Ý tưởng và thuật toán:

**procedure** bubble\_sort2(*list* L, *number* n) //n=listsize

**Fornumber *i*** from **1 to**n*-1*

**for** *number* j **from** *n*-1 **downto** i

**if** L[j] > L[j + 1] //nếu chúng không đúng thứ tự

swap(L[j], L[j + 1]) //đổi chỗ chúng cho nhau

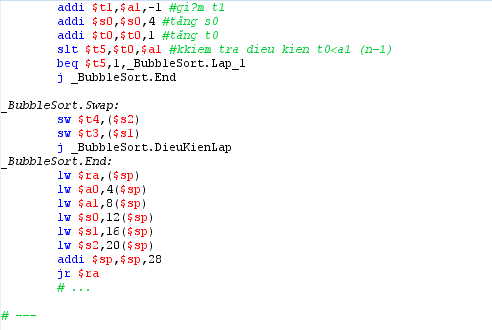
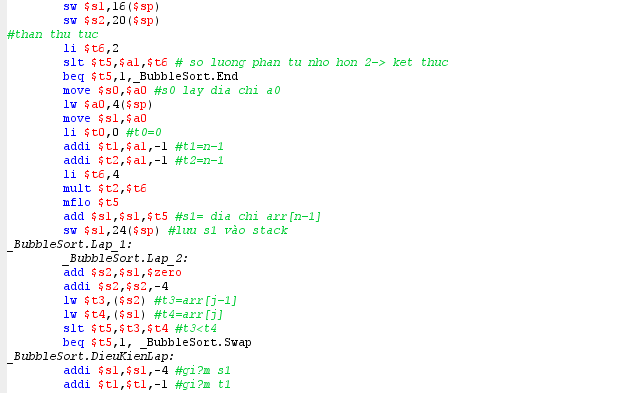
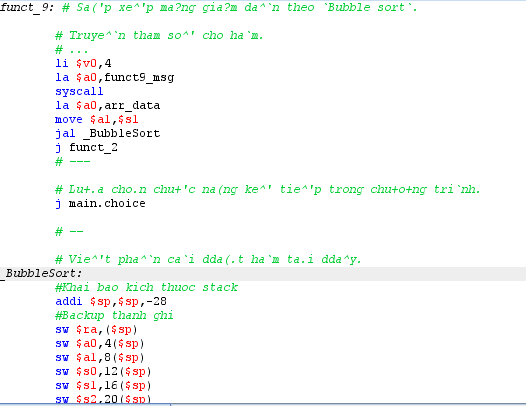
**endif**

**endfor**

**endfor**

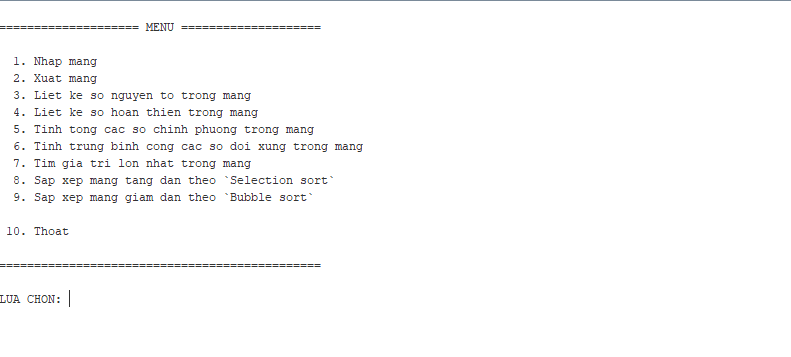
**endprocedure**

c. Cài đặt



# CẤU TRÚC XỬ LÍ CỦA CHƯƠNG TRÌNH

Trình bày dạng Menu:



Để thực hiện chức năng, nhập vào từ bàn phim số thức tự tương ứng từ 1->10 để thực hiện.

# ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH

|  |  |
| --- | --- |
| Chức năng | Mức độ hoàn thành |
| 1.Nhập mảng | 100% |
| 2.Xuất mảng | 100% |
| 3.Liệt kê các số nguyên tố | 100% |
| 4.Liệt kê các số hoàn thiện | 100% |
| 5.Tính tổng các số chính phương | 100% |
| 6. Tính TBC Các số đối xứng | 100% |
| 7.Số lớn nhất trong mảng | 100% |
| 8. Sắp xếp tăng dần Select | 100% |
| 9. Sắp xếp giảm dần Bubble | 100% |
| 10. Menu | 100% |

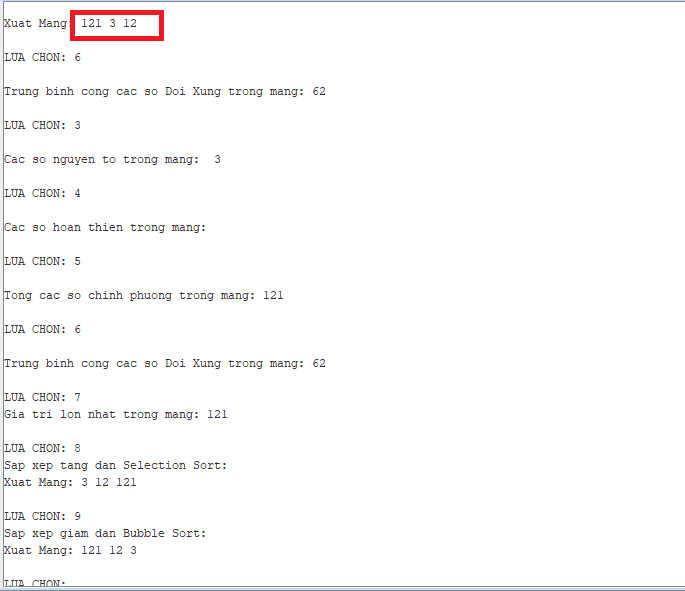
### **ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH TRÊN TOÀN BỘ CHƯƠNG TRÌNH: 100%**

## **TEST CASE**

**TEST CASE 1:**

**INPUT:**

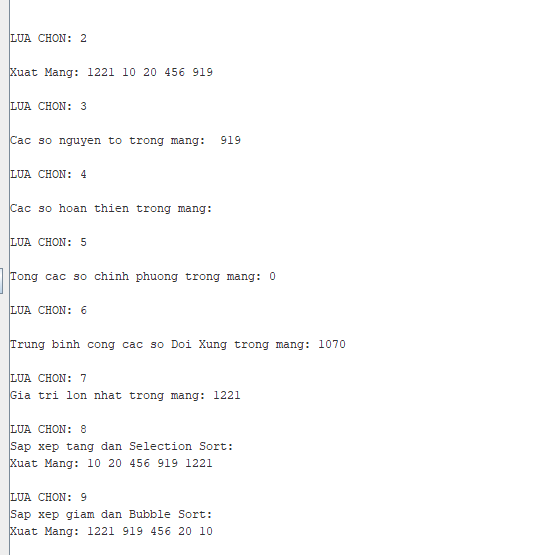
* **N: 3 phần tử**
* **Giá trị: 121 3 12**

****

**TEST CASE 2:**

**INPUT:**

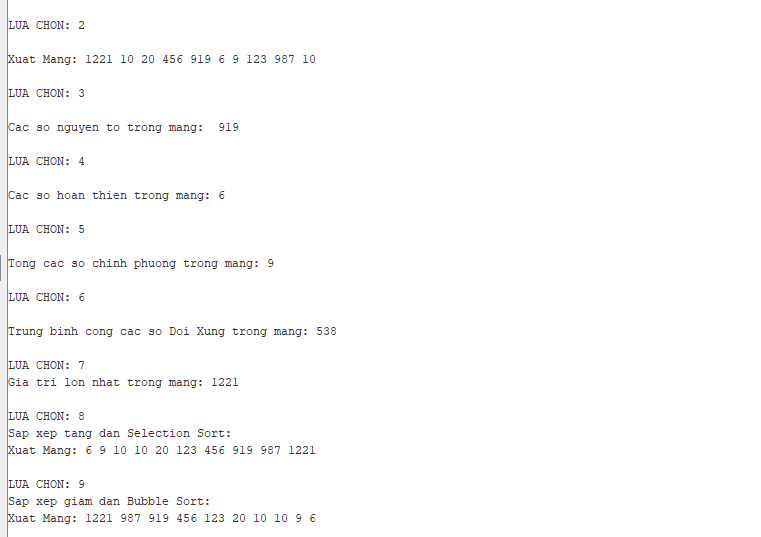
* **N: 5 phần tử**
* **Giá trị: 1221 10 20 456 919**

****

**TEST CASE 3:**

**INPUT:**

* **N: 10 phần tử**
* **Giá trị: 1221 10 20 456 919 6 9 123 987 10**

****

## **KHÓ KHĂN GẶP PHẢI**

**-**Nguồn tài liệu về đề tài quá ít.

-Chưa hoàn thiện kĩ năng làm việc nhóm.

-Thời tiết ảnh hưởng đến quá trình làm việc.

**TÀI LIỆU THAM KHẢO**

1. **Slide lý thuyết bộ môn kiến trúc máy tinh và hợp ngữ**
2. [**Sắp xếp nổi bọt (Wikipedia)**](https://vi.wikipedia.org/wiki/S%E1%BA%AFp_x%E1%BA%BFp_n%E1%BB%95i_b%E1%BB%8Dt)
3. [**http://chortle.ccsu.edu/AssemblyTutorial/index.html**](http://chortle.ccsu.edu/AssemblyTutorial/index.html)
4. [**www.scribd.com/doc/3577342/MIPS-Assembly-Language-Programming**](http://www.scribd.com/doc/3577342/MIPS-Assembly-Language-Programming)
5. [**http://dkrizanc.web.wesleyan.edu/courses/231/07/mips-spim.pdf**](http://dkrizanc.web.wesleyan.edu/courses/231/07/mips-spim.pdf)