Contents

[Bài thực hành số 1 – Tuần 7 2](#_Toc148957337)

[**Bài 1.1.**Viết một chương trình C nhập vào 3 số nguyên. Thiết lập một con trỏ để lần lượt trỏ tới từng số nguyên và hiển thị kết quả giá trị tham chiếu ngược của con trỏ. 2](#_Toc148957338)

[**Bài 1.2**. Viết chương trình in ra địa chỉ của 5 phần tử đầu tiên trong mảng được định nghĩa sau đây: int a[7]= {13, -355, 235, 47, 67, 943, 1222}; 4](#_Toc148957339)

[**Bài 1.3.** Viết chương trình yêu cầu nhập giá trị cho 3 biến số nguyên x, y, z kiểu int. Sau đó sử dụng duy nhất một con trỏ để cộng giá trị của mỗi biến thêm 100. 4](#_Toc148957340)

[**Bài 1.4.** Viết hàm countEven(int\*, int) nhận một mảng số nguyên và kích thước của mảng, trả về số lượng số chẵn trong mảng??? 7](#_Toc148957341)

[**Bài 1.5.** Viết hàm trả về con trỏ trỏ tới giá trị lớn nhất của một mảng các số double. Nếu mảng rỗng hãy trả về NULL. 8](#_Toc148957342)

[**Bài 1.6**. Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ. 10](#_Toc148957343)

[**Bài 1.7.** Viết chương trình nhập vào một mảng các số nguyên với số lượng các phần tử nhập từ bàn phím. Sau đó sắp xếp mảng theo thứ tự tăng dần. Hiển thị danh sách mảng trước và sau khi sắp xếp. 12](#_Toc148957344)

[**Bài 1.8.** Viết chương trình nhập vào một ma trận 2 chiều kích thước m\*n với m và n nhập từ bàn phím. Sau đó đưa ra tổng các phần tử chẵn của ma trận đó. 15](#_Toc148957345)

[**Bài 1.9**. Viết chương trình in ra tất cả các dãy con của một dãy cho trước. 18](#_Toc148957346)

[**Bài 1.10.** Viết chương trình nhập vào 2 ma trận vuông cùng kích thước n\*n, trong đó n nhập từ bàn phím. Sau đó tính tổng và tích của hai ma trận đó và đưa kết quả ra màn hình. Yêu cầu sử dụng cấp phát động để cấp phát bộ nhớ cho các ma trận. 19](#_Toc148957347)

**Figures**

[**Hinh 1. 1: Đề bài bài 1.1** 3](#_Toc149863475)

[**Hinh 1. 2: Code bài 1.2** 3](#_Toc149863476)

[**Hinh 1.3: Test case bài 1.3** 4](#_Toc149863477)

[**Hình 2. 1: Code và testcase bài 2** 5](#_Toc149863481)

[**Hình 3. 1: Code bài 1.3** 6](#_Toc149862601)

[**Hình 3. 2: Testcase bài 1.3** 6](#_Toc149862602)

[**Hình 4. 1: Code bài 1.4** 7](#_Toc149862607)

[**Hình 4. 2: Testcase bài 1.4** 7](#_Toc149862608)

[**Hình 5. 1: code bài 1.5** 8](#_Toc149862615)

[**Hình 5. 2: Test case bài 1.5** 8](#_Toc149862616)

[**Hình 6. 1: Code bài 1.6** 10](#_Toc149862623)

[**Hình 6. 2: Test case bài 1.6** 11](#_Toc149862624)

[**Hình 7.1: Code bài 1.7** 12](#_Toc149862630)

[**Hình 7. 2: Test case bài 1.7** 13](#_Toc149862631)

[**Hình 8. 1: Code bài 1.8** 16](#_Toc149862637)

[**Hình 8. 2: Test case bài 1.8.** 17](#_Toc149862638)

[**Hình 9 1: Code và test case bài 1.9** 20](#_Toc149862824)

[**Hinh 10. 1: Code bài 1.10** 24](#_Toc149862831)

[**Hinh 10. 2: Test case bài 1.10.** 26](#_Toc149862832)

# Bài thực hành số 1 – Tuần 7

# **Bài 1.1.**Viết một chương trình C nhập vào 3 số nguyên. Thiết lập một con trỏ để lần lượt trỏ tới từng số nguyên và hiển thị kết quả giá trị tham chiếu ngược của con trỏ.

**Lưu ý:** Phép toán & trả về địa chỉ của biến.

**Hinh 1. 1: Đề bài bài 1.1**

**Hinh 1. 2: Code bài 1.2**

**Hinh 1.3: Test case bài 1.3**

# **Bài 1.2**. Viết chương trình in ra địa chỉ của 5 phần tử đầu tiên trong mảng được định nghĩa sau đây: int a[7]= {13, -355, 235, 47, 67, 943, 1222};

**Lưu ý:  
Để in địa chỉ con trỏ các bạn sử dụng ký tự định dạng %p  
Để lấy địa chỉ của một biến ta có thể dùng phép toán &**

**Hình 2. 1: Code và testcase bài 2**

# **Bài 1.3.** Viết chương trình yêu cầu nhập giá trị cho 3 biến số nguyên x, y, z kiểu int. Sau đó sử dụng duy nhất một con trỏ để cộng giá trị của mỗi biến thêm 100.



**Hình 3. 1: Code bài 1.3**

**Hình 3. 2: Test case bài 1.3**

# **Bài 1.4.** Viết hàm countEven(int\*, int) nhận một mảng số nguyên và kích thước của mảng, trả về số lượng số chẵn trong mảng???

**Hình 4. 1: Code bài 1.4**

**Hình 4. 2: Testcase bài 1.4**

# **Bài 1.5.** Viết hàm trả về con trỏ trỏ tới giá trị lớn nhất của một mảng các số double. Nếu mảng rỗng hãy trả về NULL.

**Hình 5. 1: code bài 1.5**

**Hình 5. 2: Test case bài 1.5**

# **Bài 1.6**. Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ.

**Hình 6. 1: Code bài 1.6**

**Hình 6. 2: Test case bài 1.6**

**Bài 1.7.** Viết chương trình nhập vào một mảng các số nguyên với số lượng các phần tử nhập từ bàn phím. Sau đó sắp xếp mảng theo thứ tự tăng dần. Hiển thị danh sách mảng trước và sau khi sắp xếp.

Yêu cầu chỉ sử dụng con trỏ để truy cập mảng, không truy cập theo index mảng.

**Hình 7.1.1: Code bài 1.7**

**Hình 7. 1.2: Code bài 1.7**

**Hình 7. 2: Test case bài 1.7**

# **Bài 1.8.** Viết chương trình nhập vào một ma trận 2 chiều kích thước m\*n với m và n nhập từ bàn phím. Sau đó đưa ra tổng các phần tử chẵn của ma trận đó.

**Lưu ý:** Khi viết hàm cấp phát bộ nhớ cho một ma trận hai chiều biểu diễn bởi con trỏ int \*\*mt, nếu ta truyền con trỏ theo kiểu địa chỉ void allocate\_mem(int \*\*mt, int m, int n) sẽ dẫn tới việc cấp phát bộ nhớ cho một bản sao của con trỏ \*\*mt. Do đó, sau khi gọi hàm thì con trỏ \*\*mt gốc vẫn không được cấp phát bộ nhớ. Để cấp phát thành công cần truyền con trỏ theo dạng địa chỉ, ví dụ sử dụng con trỏ cấp 3 dạng int \*\*\*mt.

**Hình 8. 1.1: Code bài 1.8**

**Hình 8. 1.2:** **Code bài 1.8**

**Hình 8. 1.3: Code bài 1.8**

**Hình 8. 2: Test case bài 1.8.**

# **Bài 1.9**. Viết chương trình in ra tất cả các dãy con của một dãy cho trước.

**Hình 9 1: Code và test case bài 1.9**

# **Bài 1.10.** Viết chương trình nhập vào 2 ma trận vuông cùng kích thước n\*n, trong đó n nhập từ bàn phím. Sau đó tính tổng và tích của hai ma trận đó và đưa kết quả ra màn hình. Yêu cầu sử dụng cấp phát động để cấp phát bộ nhớ cho các ma trận.

**Hinh 10. 1.1: Code bài 1.10**

**Hinh 10. 1. 2: Code bài 1.10**

**Hinh 10. 2: Test case bài 1.10.**