Contents

[Bài thực hành số 1 – Tuần 32 2](#_Toc135168857)

[**Bài tập 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](#_Toc135168858)

[**Bài tập 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}; 5](#_Toc135168859)

[**Bài tập 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. 7](#_Toc135168860)

[**Bài tập 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??? 10](#_Toc135168861)

[**Bài tập 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. 12](#_Toc135168862)

[**Bài tập 1.6.** Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ. 14](#_Toc135168863)

[**Bài tập 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. 17](#_Toc135168864)

[**Bài tập 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. 21](#_Toc135168865)

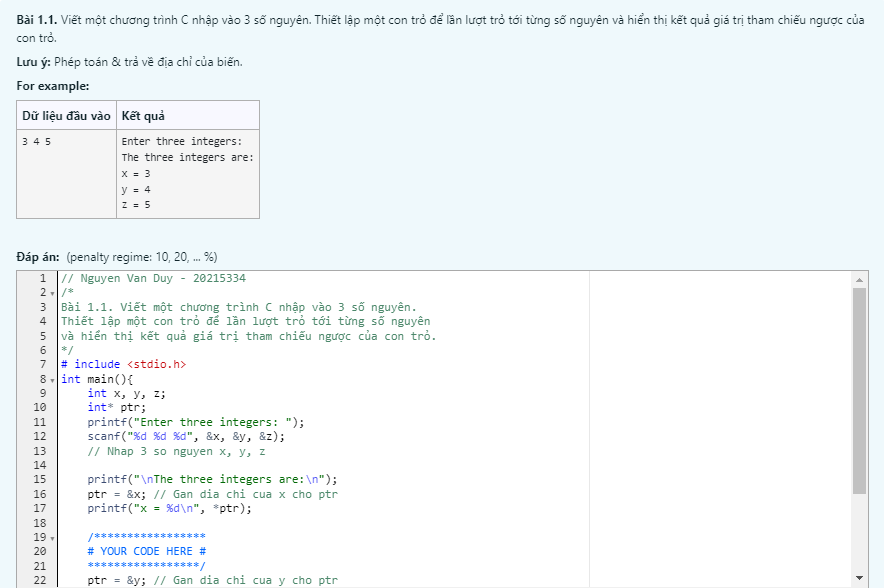
[**Bài tập 1.9.**Viết chương trình in ra tất cả các dãy con của một dãy cho trước. 27](#_Toc135168866)

[**Bài tập 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. 30](#_Toc135168867)

# Bài thực hành số 1 – Tuần 32

## **Bài tập 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.





// Nguyen Van Duy - 20215334

/\*

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.

\*/

# include <stdio.h>

int main(){

int x, y, z;

int\* ptr;

printf("Enter three integers: ");

scanf("%d %d %d", &x, &y, &z);

// Nhap 3 so nguyen x, y, z

printf("\nThe three integers are:\n");

ptr = &x; // Gan dia chi cua x cho ptr

printf("x = %d\n", \*ptr);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ptr = &y; // Gan dia chi cua y cho ptr

printf("y = %d\n", \*ptr);

ptr = &z; // Gan dia chi cua z cho ptr

printf("z = %d\n", \*ptr);

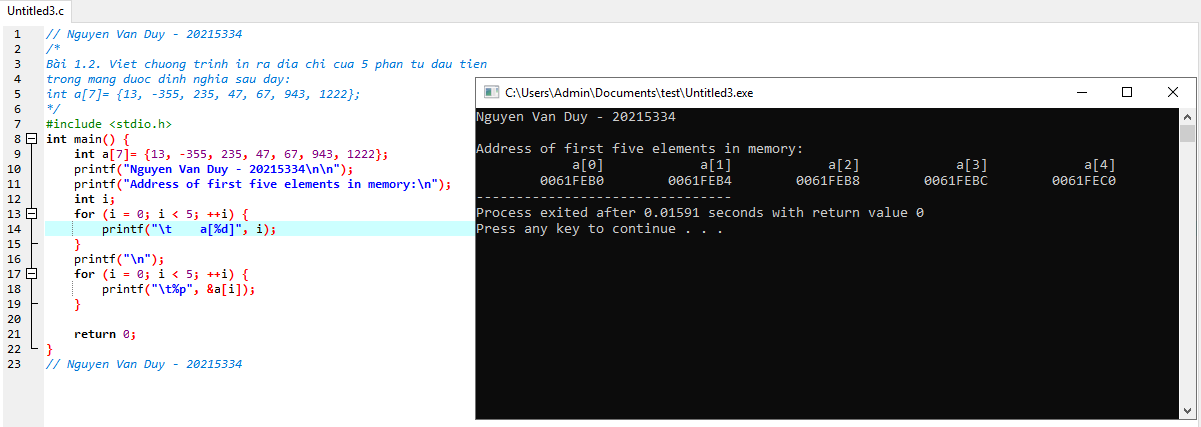
return 0;

}

// Nguyen Van Duy - 20215334

## **Bài tập 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 &**



// Nguyen Van Duy - 20215334

/\*

Bài 1.2. Viet chuong trinh in ra dia chi cua 5 phan tu dau tien

trong mang duoc dinh nghia sau day:

int a[7]= {13, -355, 235, 47, 67, 943, 1222};

\*/

#include <stdio.h>

int main() {

int a[7]= {13, -355, 235, 47, 67, 943, 1222};

printf("Nguyen Van Duy - 20215334\n\n");

printf("Address of first five elements in memory:\n");

int i;

for (i = 0; i < 5; ++i) {

printf("\t a[%d]", i);

}

printf("\n");

for (i = 0; i < 5; ++i) {

printf("\t%p", &a[i]);

}

return 0;

}

// Nguyen Van Duy – 20215334

## **Bài tập 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.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

// Nguyen Van Duy - 20215334

/\*

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.

\*/

#include <stdio.h>

int main() {

int x, y, z;

int \*ptr;

scanf("%d %d %d", &x, &y, &z);

// Nhap 3 so nguyen x, y, z

printf("Here are the values of x, y, and z:\n");

printf("%d %d %d\n", x, y, z);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ptr = &x; // Gan dia chi cua x cho ptr

\*ptr += 100; // Cong them 100

ptr = &y; // Gan dia chi cua y cho ptr

\*ptr += 100; // Cong them 100

ptr = &z; // Gan dia chi cua z cho ptr

\*ptr += 100; // Cong them 100

printf("Once again, here are the values of x, y, and z:\n");

printf("%d %d %d\n", x, y, z);

return 0;

}

// Nguyen Van Duy - 20215334

## **Bài tập 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???

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, phần mềm, hàng, số

Mô tả được tạo tự động

// Nguyen Van Duy - 20215334

/\*

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???

\*/

#include <iostream>

using namespace std;

int counteven(int\* arr, int size){

int count = 0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for (int i = 0; i < size; ++i) {

if (!(arr[i] & 1)) {

++count;

}

}

return count;

}

// Nguyen Van Duy - 20215334

## **Bài tập 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.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

// Nguyen Van Duy - 20215334

/\*

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.

\*/

#include <stdio.h>

double\* maximum(double\* a, int size){

double \*max;

max = a;

if (a==NULL) return NULL;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for (int i = 0; i < size; ++i) {

if (\*(a+i) > \*max) {

max = a+i;

}

}

return max;

}

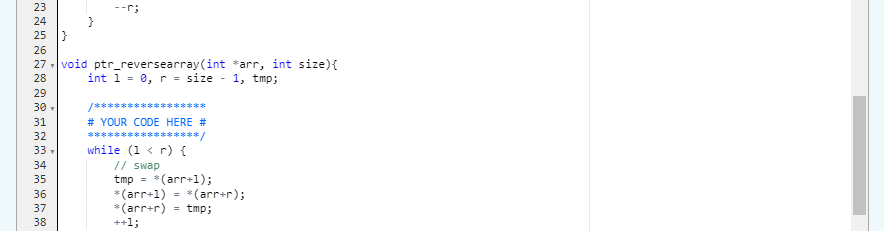
// Nguyen Van Duy - 20215334

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

Ví dụ mảng đầu vào là [9, -1, 4, 5, 7] thì kết quả là [7, 5, 4, -1, 9].

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, thiết kế

Mô tả được tạo tự động



Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

// Nguyen Van Duy - 20215334

/\*

Bài 1.6. Viết hàm đảo ngược một mảng

các số nguyên theo hai cách:

dùng chỉ số và dùng con trỏ.

\*/

#include <iostream>

using namespace std;

void reversearray(int arr[], int size){

int l = 0, r = size - 1, tmp;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while (l < r) {

// swap

tmp = arr[l];

arr[l] = arr[r];

arr[r] = tmp;

++l;

--r;

}

}

void ptr\_reversearray(int \*arr, int size){

int l = 0, r = size - 1, tmp;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while (l < r) {

// swap

tmp = \*(arr+l);

\*(arr+l) = \*(arr+r);

\*(arr+r) = tmp;

++l;

--r;

}

}

// Nguyen Van Duy – 20215334

## **Bài tập 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.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động// Nguyen Van Duy - 20215334

/\*

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.

\*/

#include <stdio.h>

#include <stdlib.h>

int \*a;

int n, tmp;

int main(){

printf("Enter the number of elements: ");

scanf("%d", &n);

//#Allocate memory

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

a = (int \*) malloc(n \* sizeof(int));

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for(int i = 0; i < n; i++)

scanf("%d", a + i);

printf("The input array is: \n");

for(int i = 0; i < n; i++)

printf("%d ", \*(a + i));

printf("\n");

//#Sort array

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for (int i = 0; i < n-1; ++i) {

for (int j = 0; j < n-i-1; ++j) {

if (\*(a+j) > \*(a+j+1)) {

// swap

int tmp = \*(a+j);

\*(a+j) = \*(a+j+1);

\*(a+j+1) = tmp;

}

}

}

printf("The sorted array is: \n");

for(int i = 0; i < n; i++)

printf("%d ", \*(a + i));

printf("\n");

free(a);

return 0;

}

// Nguyen Van Duy - 20215334

## **Bài tập 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.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, hàng, ảnh chụp màn hình, số

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

// Nguyen Van Duy - 20215334

/\*

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.

\*/

#include <stdio.h>

#include <stdlib.h>

void allocate\_mem(int \*\*\*mt, int m, int n){

//#Allocate memory for the matrix

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*mt = (int \*\*) malloc(m \* sizeof(int \*));

int i;

for (i = 0; i < m; ++i) {

\*(\*mt+i) = (int \*) malloc(n \* sizeof(int));

}

}

void input(int \*\*mt, int m, int n){

//#Input elements of the matrix

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int i, j;

for (i = 0; i < m; ++i) {

for (j = 0; j < n; ++j) {

printf("mt[%d][%d] = ", i, j);

scanf("%d", \*(mt + i) + j);

}

}

}

void output(int \*\*mt, int m, int n){

//# Print all elements of the matrix

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int i, j;

for (i = 0; i < m; ++i) {

for (j = 0; j < n; ++j) {

printf("%d ", \*(\*(mt + i) + j));

}

printf("\n");

}

}

int process(int \*\*mt, int m, int n){

int tong = 0;

//# Calculate the sum of all even elements in the matrix

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int i, j;

for (i = 0; i < m; ++i) {

for (j = 0; j < n; ++j) {

if (!(\*(\*(mt + i) + j) & 1)) {

tong += \*(\*(mt + i) + j);

}

}

}

return tong;

}

void free\_mem(int \*\*mt, int m, int n){

//# Free memory

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# YOUR CODE HERE #

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int i;

for (i = 0; i < m; ++i) {

free(mt[i]);

}

free(mt);

}

int main(){

int m, n, \*\*mt;

printf("Enter m, n = ");

scanf("%d%d", &m, &n);

allocate\_mem(&mt, m, n);

input(mt, m, n);

output(mt, m, n);

printf("The sum of all even elements is %d", process(mt, m, n));

free\_mem(mt, m, n);

return 0;

}

// Nguyen Van Duy - 20215334

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

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

// Nguyen Van Duy - 20215334

/\*

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

\*/

#include <iostream>

using namespace std;

void process(int \*a, int n) {

for (int i = 0; i < n; ++i) {

string s;

for (int j = i; j < n; ++j) {

s += to\_string(a[j]) + " ";

cout << s << endl;

}

}

}

int main() {

int \*a, n;

// nhap gia tri cho n

cin >> n;

// Allocate memory for the array

a = new int[n];

// nhap gia tri cho mang

for (int i = 0; i < n; ++i) {

cin >> a[i];

}

process(a, n);

// free memory

delete[] a;

}

// Nguyen Van Duy – 20215334

## **Bài tập 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.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, máy tính

Mô tả được tạo tự động

// Nguyen Van Duy - 20215334

/\*

\* 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.

\*/

#include <iostream>

using namespace std;

void allocate(int \*\*\*mt, int n) {

// Allocate memory for the matrix

\*mt = new int \* [n];

for (int i = 0; i < n; ++i) {

(\*mt)[i] = new int[n];

}

}

void input(int \*\*mt, int n) {

// nhap gia tri cho ma tran

for (int i = 0; i < n; ++i) {

for (int j = 0; j < n; ++j) {

cin >> mt[i][j];

}

}

}

void sum(int \*\*mtA, int \*\*mtB, int n) {

for (int i = 0; i < n; ++i) {

for (int j = 0; j < n; ++j) {

cout << mtA[i][j] + mtB[i][j] << " ";

}

cout << endl;

}

}

void mul(int \*\*mtA, int \*\*mtB, int n) {

for (int i = 0; i < n; ++i) {

for (int j = 0; j < n; ++j) {

int val = 0;

for (int k = 0; k < n; ++k) {

val += mtA[i][k] \* mtB[k][j];

}

cout << val << " ";

}

cout << endl;

}

}

void free(int \*\*mt, int n){

// Free memory

for (int i = 0; i < n; ++i) {

delete[] mt[i];

}

if (n > 0) {

delete[] mt;

}

}

int main() {

int \*\*mtA, \*\*mtB, n;

// nhap gia tri cho n

cin >> n;

// Allocate memory

allocate(&mtA, n);

allocate(&mtB, n);

input(mtA, n);

input(mtB, n);

// output

sum(mtA, mtB, n);

mul(mtA, mtB, n);

// free memory

free(mtA, n);

free(mtB, n);

}

// Nguyen Van Duy - 20215334