**Contents**

[Danh mục hình ảnh 1](#_heading=h.gjdgxs)

[Bài thực hành số 1: Con trỏ và cấp phát](#_heading=h.30j0zll) 3

[Phần 1. Thực hành về con trỏ](#_heading=h.1fob9te) 3

[1.](#_heading=h.3znysh7) Bài tập 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. 3

[2. Bài tập 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 &](#_heading=h.tyjcwt) 5

[3. Bài tập 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](#_heading=h.3dy6vkm) 6

[Phần 2. Con trỏ và mảng](#_heading=h.1t3h5sf) 8

[1. Bài tập 4: Viết hàm countEven(int\*, int) nhận một mảng số nguyên và kích thước của mảng, trả về số lượng số chẵn trong mảng.](#_heading=h.4d34og8) 8

[2. Bài tập 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.](#_heading=h.2s8eyo1) 9

[3. Bài tập 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].](#_heading=h.17dp8vu) 11

[Phần 3. Cấp phát động](#_heading=h.3rdcrjn) 14

[1. Bài tập 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.](#_heading=h.26in1rg) 14

[2. Bài tập 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(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](#_heading=h.lnxbz9) 16

[Phần 3. Bài tập về nhà](#_heading=h.35nkun2) 20

[1. Bài tập 9: Viết chương trình in ra tất cả các dãy con của một dãy cho trước](#_heading=h.1ksv4uv) 20

[2. Bài tập 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.](#_heading=h.44sinio) 22

# Danh mục hình ảnh

[Hình 1 Bài 1.1 Phép toán & trả về địa chỉ của biến 3](#_Toc55587270)

[Hình 2 Bài 1.2 In ra địa chỉ của các phần tử trong mảng 5](#_Toc55587271)

[Hình 3 Bài 1.3 In ra giá trị các phần tử khi cộng thêm 100. 6](#_Toc55587272)

[Hình 4 Bài 1.4 Trả về số lượng số chẵn trong mảng. 8](#_Toc55587273)

[Hình 5 Bài 1.5 Trả về con trỏ trỏ tới phần tử lớn nhất trong mảng. 9](#_Toc55587274)

[Hình 6 Bài 1.6 Đảo ngược các số trong mảng. 12](#_Toc55587275)

[Hình 7 Bài 1.7 Hiển thị danh sách mảng trước và sau khi sắp xếp. 15](#_Toc55587276)

[Hình 8 Bài 1.8 Nhập và xuất một mảng. 19](#_Toc55587277)

[Hình 9 Bài 1.9 In ra tất cả các tập con. 20](#_Toc55587278)

[Hình 10 Bài 1.10 In ra tổng và tích của 2 ma trận. 23](#_Toc55587279)

# Bài thực hành số 1: Con trỏ và cấp phát

## Phần 1. Thực hành về con trỏ

### Bài tập 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.

**Tên file: 20183554-NguyenQuangHuy\_Bai 1\_1.cpp**

# include <stdio.h>

int main(){

int x, y, z;

int\* ptr;

printf("Ho Va Ten: Nguyen Quang Huy\n");

printf("MSSV: 20183554\n\n");

printf("Enter three integers: ");

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

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

ptr = &x;

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

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

ptr = &y;

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

ptr = &z;

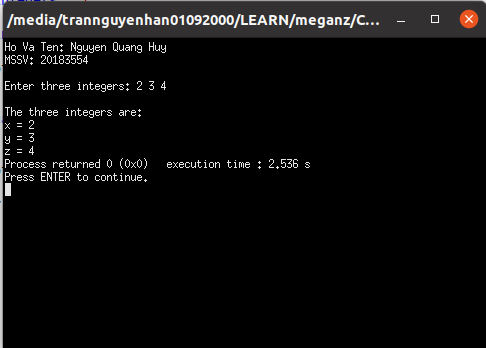
printf("z = %d", \*ptr);

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

return 0;

}

Kết quả:



Hình Bài 1.1 Phép toán & trả về địa chỉ của biến

### Bài tập 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 &

### Tên file: 20183554-NguyenQuangHuy\_Bai 1\_2.cpp

### #include <stdio.h>

### int main(){

### printf("Ho Va Ten: Nguyen Quang Huy\n");

### printf("MSSV: 20183554\n\n");

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

### printf("address of first five elements in memory.\n");

### for (int i=0; i<5;i++) printf("\ta[%d] ",i);

### printf("\n");

### 

### /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Ho va ten : Nguyen Quang Huy

### MSSV : 20183554 \*/

### for (int i=0; i<5;i++) printf("\t%p ",&a[i]);

### printf("\n");

### /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

### 

### return 0;

### }

### Kết quả:

### 

Hình Bài 1.2 In ra địa chỉ của các phần tử trong mảng

### 

### Bài tập 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

### Tên file: 20183554-NguyenQuangHuy\_Bai 1\_3.cpp

### #include <stdio.h> int main()

{

printf("Ho Va Ten: Nguyen Quang Huy\n");

printf("MSSV: 20183554\n\n");

int x, y, z;

int \*ptr;

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

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

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

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

ptr = &x;

\*ptr = \*ptr+100;

ptr = &y;

\*ptr = \*ptr+100;

ptr = &z;

\*ptr = \*ptr+100;

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

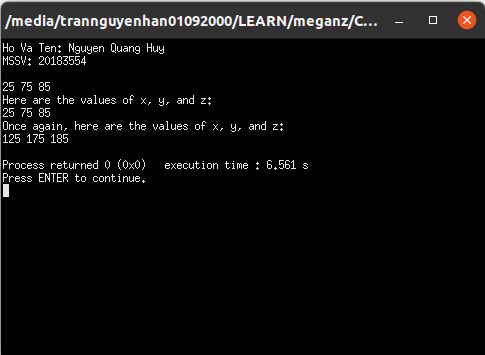
printf("Once again, here are the values of x, y, and z:\n");

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

return 0;

}

### Kết quả:



Hình Bài 1.3 In ra giá trị các phần tử khi cộng thêm 100.

## Phần 2. Con trỏ và mảng

### Bài tập 4: Viết hàm countEven(int\*, int) nhận một mảng số nguyên và kích thước của mảng, trả về số lượng số chẵn trong mảng.

### Tên file: 20183554-NguyenQuangHuy\_Bai 1\_4.cpp

### #include<bits/stdc++.h>

using namespace std;

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

int count = 0;

printf("Ho Va Ten: Nguyen Quang Huy\n");

printf("MSSV: 20183554\n\n");

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

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

if(arr[i]%2==0) count++;

}

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

return count;

}

int main(){

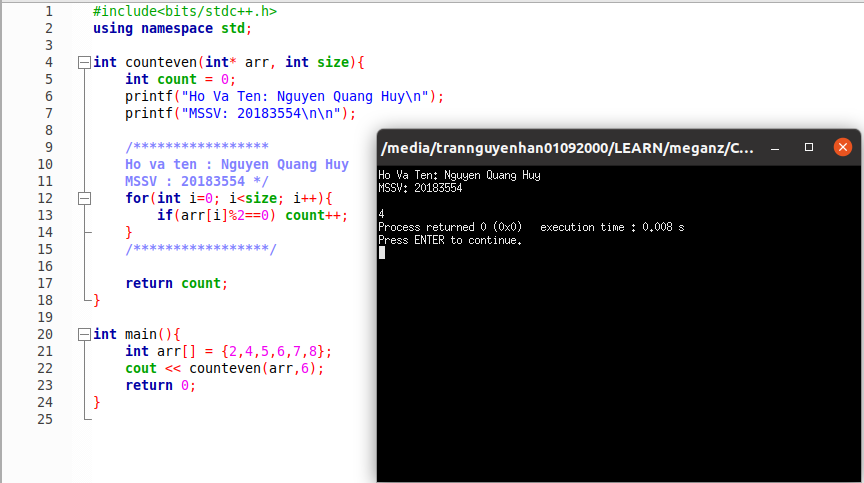
int arr[] = {2,4,5,6,7,8};

cout << counteven(arr,6);

return 0;

}

Kết quả :



Hình Bài 1.4 Trả về số lượng số chẵn trong mảng.

### Bài tập 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.

### Tên file: 20183554-NguyenQuangHuy\_Bai 1\_5.cpp

### #include<bits/stdc++.h>

using namespace std;

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

double \*max;

max = a;

if (a==NULL) return NULL;

printf("Ho Va Ten: Nguyen Quang Huy\n");

printf("MSSV: 20183554\n\n");

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

for(int i=1; i<size; i++){

if(a[i] > \*max) max = &a[i];

}

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

return max;

}

int main(){

double arr[] = {2,4,9,6,7,8};

cout << \*maximum(arr,6);

return 0;

}

Kết quả:

## 

Hình Bài 1.5 Trả về con trỏ trỏ tới phần tử lớn nhất trong mảng.

### Bài tập 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].

### Tên file: 20183554-NguyenQuangHuy\_Bai 1\_6.cpp

### #include<bits/stdc++.h>

using namespace std;

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

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

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

while(l<r){

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;

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

while(l<r){

tmp = arr[l];

arr[l] = arr[r];

arr[r] = tmp;

l++;

r--;

}

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

}

int main(){

printf("Ho Va Ten: Nguyen Quang Huy\n");

printf("MSSV: 20183554\n\n");

int arr[] = {2,4,9,6,7,8};

ptr\_reversearray(arr,6);

for(int i=0; i<6; i++)

cout << arr[i] << " ";

}

Đáp án:



Hình Bài 1.6 Đảo ngược các số trong mảng.

## Phần 3. Cấp phát động

### Bài tập 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.

**Tên file: 20183554-NguyenQuangHuy\_Bai 1\_7.cpp**

#include <stdio.h>

int \*a;

int n, tmp;

int main(){

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

scanf("%d", &n);

//#Allocate memory

printf("Ho Va Ten: Nguyen Quang Huy\n");

printf("MSSV: 20183554\n\n");

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

a = new int[n];

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

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

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

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

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

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

int tmp;

tmp = \*(a+i);

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

\*(a+j) = tmp;

}

}

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

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

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

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

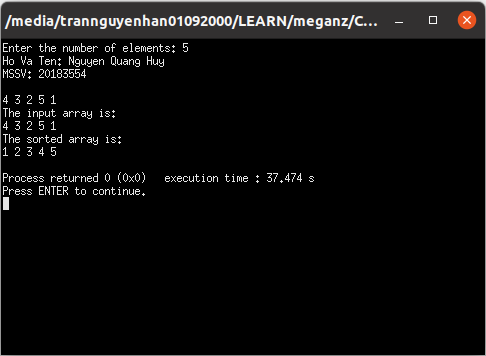
printf("\n");

delete [] a;

return 0;

}

Đáp án:



Hình Bài 1.7 Hiển thị danh sách mảng trước và sau khi sắp xếp.

### Bài tập 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(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

**Tên file: 20183554-NguyenQuangHuy\_Bai 1\_8.cpp**

#include <stdio.h>

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

//#Allocate memory for the matrix

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

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

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

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

}

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

}

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

//#Input elements of the matrix

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

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

for(int 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

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

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

for(int 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

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

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

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

if(mt[i][j]%2==0)

tong += mt[i][j];

}

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

return tong;

}

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

//# Free memory

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

Ho va ten : Nguyen Quang Huy

MSSV : 20183554 \*/

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

delete []mt[i];

delete []mt;

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

}

int main(){

printf("Ho Va Ten: Nguyen Quang Huy\n");

printf("MSSV: 20183554\n\n");

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);

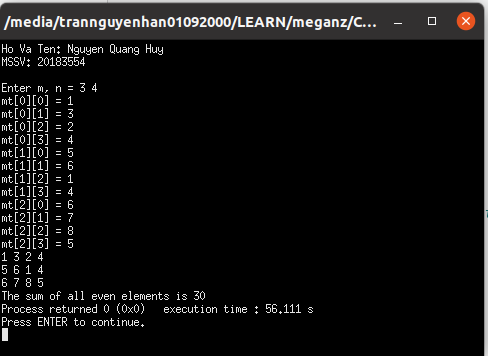
printf("The sum of all even elements is %d", process(mt, m, n));

free\_mem(mt, m, n);

return 0;

}

Kết quả:



Hình Bài 1.8 Nhập và xuất một mảng.

## Phần 3. Bài tập về nhà

### Bài tập 9: Viết chương trình in ra tất cả các dãy con của một dãy cho trước. Ví dụ dãy 1 3 4 2 có các dãy con sau: 1 1 3 1 3 4 1 3 4 2 3 3 4 3 4 2 4 4 2 2

**Tên file: 20183554-NguyenQuangHuy\_Bai 1\_9.cpp**

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;

cin >> n;

int arr[n];

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

cin >> arr[i];

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

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

for(int k=i; k<=j; k++)

cout << arr[k] << " ";

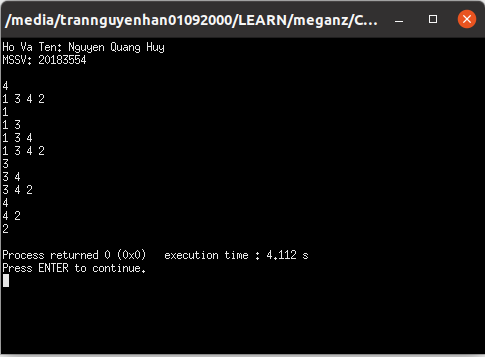
cout << endl;

}

return 0;

}

Kết quả:



Hình Bài 1.9 In ra tất cả các tập con.

### Bài tập 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.

**Tên file: 20183554-NguyenQuangHuy\_Bai 1\_10.cpp**

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;

int \*\*mt1, \*\*mt2;

cin >> n;

// Cap phat bo nho cho mt1, mt2

mt1 = new int\*[n];

mt2 = new int\*[n];

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

mt1[i] = new int[n];

mt2[i] = new int[n];

}

// Nhap gia tri cho mt1, mt2

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

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

cin >> mt1[i][j];

}

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

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

cin >> mt2[i][j];

}

// Tinh va in va tong cua 2 ma tran

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

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

int temp;

temp = mt1[i][j] + mt2[i][j];

cout << temp << " ";

}

cout << endl;

}

// Tinh va in ra tich cua 2 ma tran

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

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

int temp = 0;

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

temp += mt1[i][k]\*mt2[k][j];

}

cout << temp << " ";

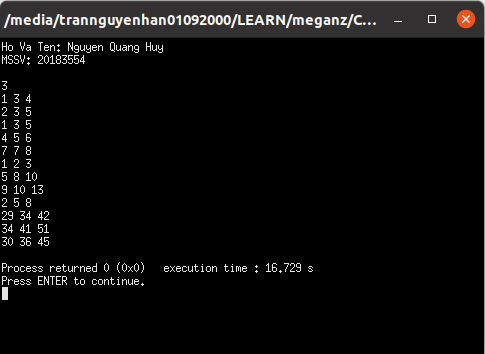
}

cout << endl;

}

}

Kết quả:



Hình Bài 1.10 In ra tổng và tích của 2 ma trận.