**BÁO CÁO BUỔI THỰC HÀNH SỐ 1** – Tuần 08

Bộ Môn Kỹ Thuật Lập Trình

A blue logo with black background

Description automatically generated

A red and white logo

Description automatically generated

Sinh viên: **Nguyễn Duy Khánh**

MSSV: **20225019**

Mã lớp thực hành: **744469**

Giáo viên hướng dẫn: **ThS. Lê Thị Hoa**

Mục lục nội dung

[Bài thực hành số 01 – Tuần 8 4](#_Toc161238431)

[BÀI TẬP TRÊN LAP 4](#_Toc161238432)

[**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. 4](#_Toc161238433)

[**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](#_Toc161238434)

[**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. 6](#_Toc161238435)

[**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. 9](#_Toc161238436)

[**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](#_Toc161238437)

[**Bài tập 1.6.** Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ. 13](#_Toc161238438)

[**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. 16](#_Toc161238439)

[**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. 20](#_Toc161238440)

[**BÀI TẬP VỀ NHÀ** 23](#_Toc161238441)

[**Bài tập 1.9.** Viết chương trình in ra tất cả các dãy con của một dãy cho trước. 23](#_Toc161238442)

[**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. 27](#_Toc161238443)

Mục lục hình ảnh

[Hình 1 Bài 1.1 In ra giá trị tham chiếu ngược của con trỏ 4](#_Toc161238452)

[Hình 2 Bài 1.2 In ra địa chỉ của phần tử trong mảng 6](#_Toc161238453)

[Hình 3 Bài 1.3 Dùng con trỏ cộng giá trị mỗi biến thêm 100 8](#_Toc161238454)

[Hình 4 Bài 1.4 Hàm trả về số phần tử chẵn trong mảng 10](#_Toc161238455)

[Hình 5 Bài 1.5 Dùng con trỏ tìm phần tử lớn nhất trong mảng 12](#_Toc161238456)

[Hình 6 Bài 1.6 Hàm đảo ngược các phần tử trong mảng dùng chỉ số và con trỏ 14](#_Toc161238457)

[Hình 7 Bài 1.7 Dùng con trỏ sắp xếp mảng theo thứ tự tăng dần 17](#_Toc161238458)

[Hình 8 Bài 1.8 Nhập mảng 2 chiều rồi tính tổng các phần tử chẵn 19](#_Toc161238459)

# Bài thực hành số 01 – Tuần 08

# BÀI TẬP TRÊN LAP

## **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.

A screenshot of a computer

Description automatically generated

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

Mô tả được tạo tự động

Hình 1 - Bài 1.1 In ra giá trị tham chiếu ngược của con trỏ

/\*

Bai 1.1 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

# include <stdio.h>

int main(){

    int x, y, z;

    int\* ptr;

// Nhap vao 3 so nguyen x, y, z

    printf("Enter three integers: ");

    scanf("%d %d %d", &x, &y, &z);

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

    ptr = &x;

    printf("x = %d\n", \*ptr); // Gan dia chi cua bien x cho con tro ptr

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    // Nguyen Duy Khanh - 20225019

// Ma lop TH: 744469

    ptr = &y;

    printf("y = %d\n", \*ptr); // Gan dia chi cua bien y cho con tro ptr

    ptr = &z;

    printf("z = %d\n", \*ptr); // Gan dia chi cua bien z cho con tro ptr

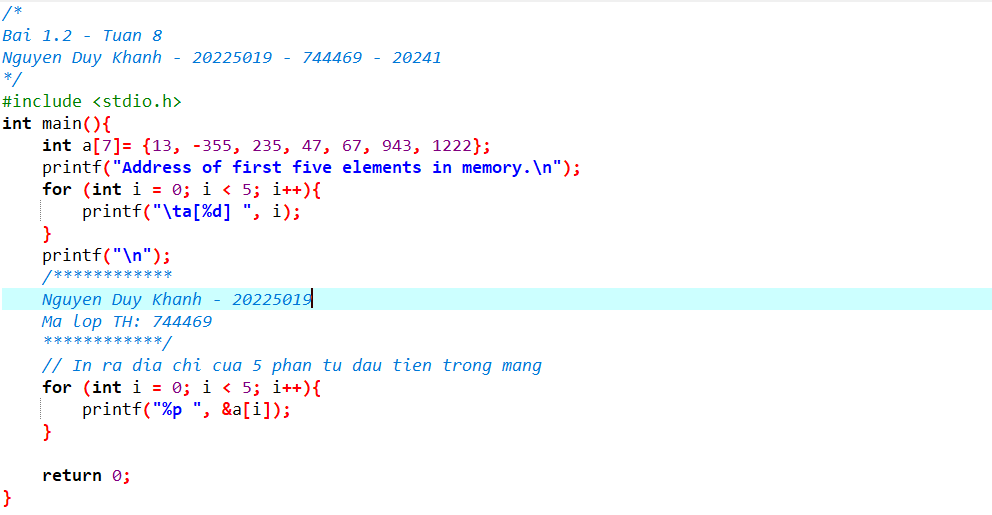
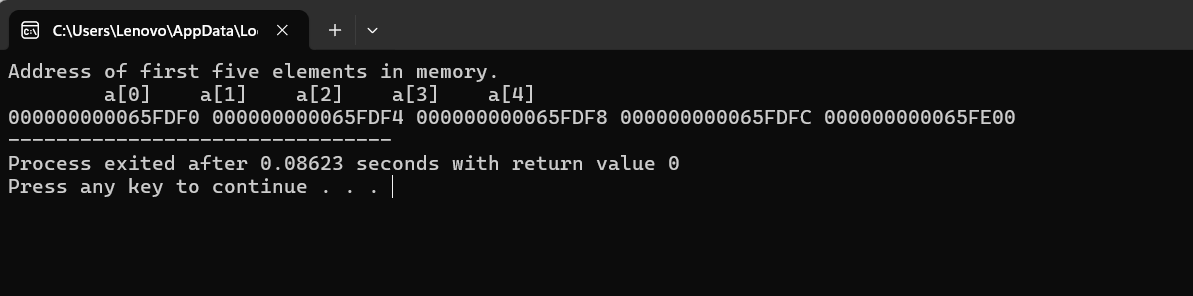
    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    return 0;

}

## **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 &

* *

Hình 2 - Bài 1.2 In ra địa chỉ của phần tử trong mảng

/\*

Bai 1.2 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include <stdio.h>

int main(){

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");

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

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

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

// In ra dia chi cua 5 phan tu dau tien trong mang

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

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

}

return 0;

}

## **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.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 3 Bài 1.3 Dùng con trỏ cộng giá trị mỗi biến thêm 100

/\*

Bai 1.3 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include <stdio.h>

int main()

{

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);

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

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

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

ptr = &x;

// Add 100 to x

\*ptr = \*ptr + 100;

ptr = &y;

// Add 100 to y

\*ptr = \*ptr + 100;

ptr = &z;

// Add 100 to 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;

}

## **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.

A screenshot of a computer program

Description automatically generated

Hình 4: Bài 1.4 Hàm trả về số phần tử chẵn trong mảng

/\*

Bai 1.4 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<stdio.h>

#include<stdlib.h>

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

int count = 0;

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

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469 - 20241

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

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

if (\*(arr + i) % 2 == 0){

count += 1;

}

}

return count;

}

void solve(){

int arr[] = {1, 5, 4, 8, 10, 6, 7, 2};

printf("%d", counteven(arr, sizeof(arr) / sizeof(arr[0])));

}

int main(){

solve();

return 0;

}

## **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.

A screenshot of a computer program

Description automatically generated

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

Mô tả được tạo tự động

Hình 5 Bài 1.5 Dùng con trỏ tìm phần tử lớn nhất trong mảng

#include<stdio.h>

#include<stdlib.h>

/\*

Bai 1.5 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

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

double \*max;

max = a;

if (a == NULL) return NULL;

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

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469 - 20241

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

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

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

// \*max = \*(a + i);

max = &a[i];

}

}

return max;

}

int main(){

double arr[] = {1., 10., 2., -7., 25., 3.};

double \*max = maximum(arr, 6);

printf("%.0f", \*max);

return 0;

}

## **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].**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Hình 6 Bài 1.6 Hàm đảo ngược các phần tử trong mảng dùng chỉ số và con trỏ

#include<stdio.h>

#include<stdlib.h>

/\*

Bai 1.6 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

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

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

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;

while (l <= r){

tmp = \*(arr + l);

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

\*(arr + r) = tmp;

l++;

r--;

}

}

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

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469 - 20241

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

int main(int argc, const char \*\* argv){

int arr[] = {9, 3, 5, 6, 2, 5};

reversearray(arr, sizeof(arr) / sizeof(arr[0]));

for (int i = 0; i < sizeof(arr) / sizeof(arr[0]); i++) {

printf("%d ", arr[i]);

}

printf("\n");

int arr2[] = {4, -1, 5, 9};

ptr\_reversearray(arr2, sizeof(arr2) / sizeof(arr2[0]));

for (int i = 0; i < sizeof(arr2) / sizeof(arr2[0]); i++) {

printf("%d ", arr2[i]);

}

}

## 

## **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.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

Mô tả được tạo tự động

Hình 7 Bài 1.7 Dùng con trỏ sắp xếp mảng theo thứ tự tăng dần

/\*

Bai 1.7 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<stdio.h>

void solve(){

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

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469 - 20241

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

int n;

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

scanf("%d", &n);

int arr[n + 1];

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

scanf("%d", &arr[i]);

}

// Print elements in this array:

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

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

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

}

// Swap elements in array:

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

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

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

int tmp = \*(arr + i);

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

\*(arr + j) = tmp;

}

}

}

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

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

printf("%d ", arr[i]);

}

}

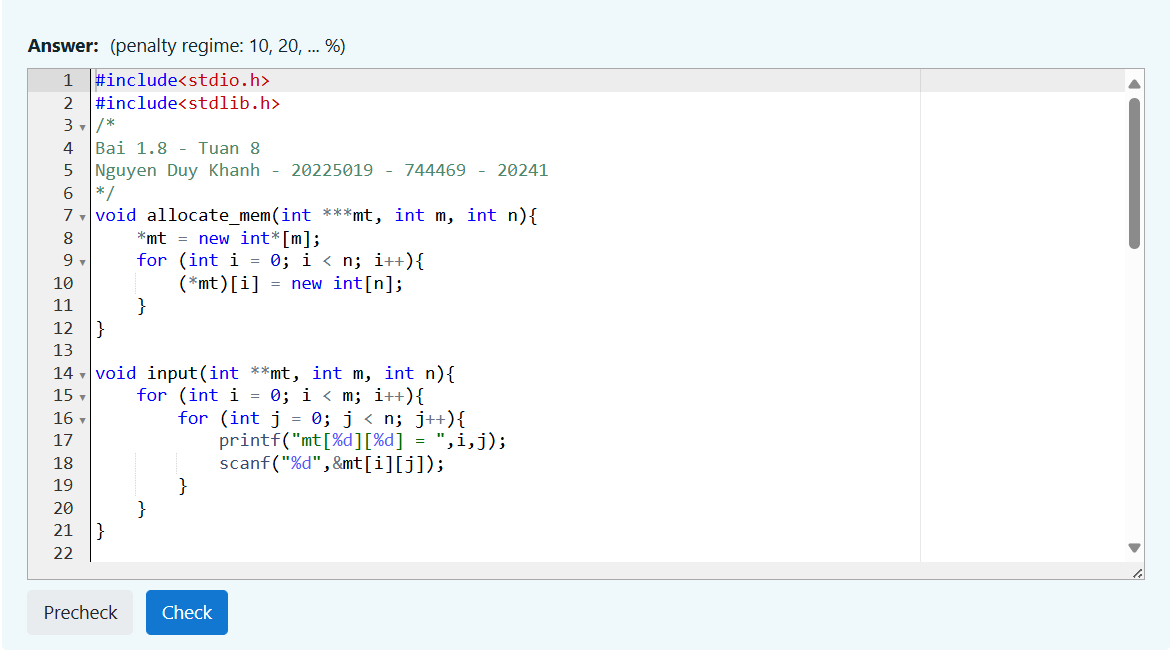
int main(){

solve();

return 0;

}

## **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.



A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 8 Bài 1.8 Nhập mảng 2 chiều rồi tính tổng các phần tử chẵn

#include<stdio.h>

#include<stdlib.h>

/\*

Bai 1.8 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

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

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

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

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

}

}

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

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){

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;

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){

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

delete[] mt[i];

}

delete[] mt;

}

int main(){

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

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469 - 20241

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

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;

}

# **BÀI TẬP VỀ NHÀ**

Chụp ảnh kết quả của tất cả các test.

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

A screenshot of a computer code

Description automatically generated

A computer screen shot of a code

Description automatically generated

A computer screen with white text

Description automatically generated

A screenshot of a computer program

Description automatically generated

/\*

Bai 1.9 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<iostream>

#include<stdlib.h>

int n;

int \*a;

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

\*a = new int[n];

}

// Allocation

void input(){

std::cin>>n;

allocateMem(&a, n);

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

std::cin>>a[i];

}

}

void freeMem(int \*a){

delete[] a;

}

// Print subarray from index start to index end

void printSubarray(int \*a, int start, int end){

for (int i = start; i <= end; i++){

std::cout<<a[i]<<" ";

}

std::cout<<std::endl;

}

int main(int argc, const char\*\* argv){

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

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

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

input();

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

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

printSubarray(a, i, j);

}

}

freeMem(a);

}

## **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.

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

A white screen with red and blue text

Description automatically generated

A computer screen with white text

Description automatically generated

A computer screen with white text

Description automatically generated

/\*

Bai 1.10 - Tuan 8

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include <bits/stdc++.h>

using namespace std;

int n;

int \*\*a1, \*\*a2, \*\*res;

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

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

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

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

}

}

void input() {

cin >> n;

allocateMem(&a1, n);

allocateMem(&a2, n);

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

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

cin >> a1[i][j];

}

}

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

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

cin >> a2[i][j];

}

}

}

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

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

delete[] a[i];

}

delete[] a;

}

int\*\* sumMt(int \*\*a1, int \*\*a2, int n) {

int \*\*res;

allocateMem(&res, n);

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

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

res[i][j] = a1[i][j] + a2[i][j];

}

}

return res;

}

int\*\* mulMt(int \*\*a1, int \*\*a2, int n) {

int \*\*res;

allocateMem(&res, n);

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

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

res[i][j] = 0;

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

res[i][j] += a1[i][k] \* a2[k][j];

}

}

}

return res;

}

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

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

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

cout << a[i][j] << " ";

}

cout << endl;

}

}

int main() {

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

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469 - 20241

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

input();

res = sumMt(a1, a2, n);

printArr(res, n);

freeMem(res, n);

res = mulMt(a1, a2, n);

printArr(res, n);

freeMem(res, n);

freeMem(a1, n);

freeMem(a2, n);

}