**CÁCH ĐÁNH GIÁ ĐIỂM THỰC HÀNH**

**HỌC PHẦN: IT3040 - KỸ THUẬT LẬP TRÌNH – 2023.1**

1. **Quy định, yêu cầu:** 
   * Tài liệu và nội dung thực hành chấm điểm trên hệ thống:

https://lab.soict.hust.edu.vn/

* + Bài tập trên lớp chấm điểm tự động (các bài không chấm trên hệ thống làm vào máy tính 🡺 làm báo cáo thực hành – Theo mẫu).
  + Hạn nộp báo cáo trên Teams (Bài tập trên lớp + Bài tập về nhà): 1 tuần.

1. **Đánh giá điểm thực hành**

1. Chuyên cần (đúng giờ, nghiêm túc trong giờ học) - Điểm danh trên Teams: 10%

2. Báo cáo thực hành (bài tập trên lớp + Về nhà) theo mẫu nộp trên Teams: 40%

3. Trắc nghiệm – Form trên Teams: 10%

4. Kiểm tra thực hành: 40%. (Tiết 2,3 buổi thực hành thứ 5).

**Điểm thưởng: 5% 🡪 10% (Cho Mục 1,2 điểm TB từ 9-10).**

Tham gia thực hành đúng giờ đầy đủ theo thời khóa biểu (nếu có lý do không đi thực hành đúng kíp được thì gửi mail xin phép thực hành bù trước 1 ngày qua mail [hoalt@soict.hust.edu.vn](mailto:hoalt@soict.hust.edu.vn), Tiêu đề: đăng ký học bù – IT3040 – MaLopTH.

Các kíp có thể bù:

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Thời gian, địa điểm, Tuần học** | **Mã nhóm** | **Mã lớp** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |
| **5** |  |  |  |
| **6** |  |  |  |
| **7** |  |  |  |

**Nếu nghỉ không có lý do 3 buổi, không thực hành bù thì điểm chuyên cần, báo cáo và BTVN coi như 0 điểm thực hành.**

Mục lục

[Bài thực hành số 1 4](#_Toc149849746)

[**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. 4](#_Toc149849747)

[**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}; 6](#_Toc149849748)

[**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](#_Toc149849749)

[**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](#_Toc149849750)

[**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](#_Toc149849751)

[**Bài tập 1.6**: Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ. 14](#_Toc149849752)

[**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](#_Toc149849753)

[**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](#_Toc149849754)

[**Bài tập 1.9**: 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: 25](#_Toc149849755)

[**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. 35](#_Toc149849756)

Danh mục hình ảnh

[Hình 1 bài 1.1 4](#_Toc149837981)

[Hình 2 Kết quả bài 1.1 4](#_Toc149837982)

[Hình 3 bài 1.2 và kết quả 6](#_Toc149837983)

[Hình 4 bài 1.3 7](#_Toc149837984)

[Hình 5 Kết quả bài 1.3 8](#_Toc149837985)

[Hình 6 bài 1.4 10](#_Toc149837986)

[Hình 7 Kết quả bài 1.4 10](#_Toc149837987)

[Hình 8 bài 1.5 12](#_Toc149837988)

[Hình 9 kết quả bài 1.5 12](#_Toc149837989)

[Hình 10 bài 1.6 14](#_Toc149837990)

[Hình 11 Kết quả bài 1.6 15](#_Toc149837991)

[Hình 12 bài 1.7 17](#_Toc149837992)

[Hình 13 Kết quả bài 1.7 18](#_Toc149837993)

[Hình 14 bài 1.8 và kết quả 22](#_Toc149837994)

[Hình 15 bài 1.9 26](#_Toc149837995)

[Hình 16 bài 1.9 test case 1 27](#_Toc149837996)

[Hình 17 bài 1.9 test case 2 27](#_Toc149837997)

[Hình 18 Bài 1.9 test case 3 27](#_Toc149837998)

[Hình 19 bài 1.9 test case 4 29](#_Toc149837999)

[Hình 20 bài 1.9 test case 5 34](#_Toc149838000)

[Hình 21 bài 1.10 37](#_Toc149838001)

[Hình 22 bài 1.10 test case 1 38](#_Toc149838002)

[Hình 23 bài 1.10 test case 2 38](#_Toc149838003)

[Hình 24 bài 1.10 test case 3 39](#_Toc149838004)

[Hình 25 bài 1.10 test case 4 41](#_Toc149838005)

Bài thực hành số 1

**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.

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

Mô tả được tạo tự động

Hình bài 1.1

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

Mô tả được tạo tự động

Hình Kết quả bài 1.1

/\*Tô Thái Linh 20215414

Mã lớp thực hành 732826\*/

#include <stdio.h>

int main() {

int x, y, z; //Khai bao 3 bien kieu int

int\* ptr; //Khai bao con tro kieu int

printf("Enter three integers:");

scanf("%d %d %d", &x, &y, &z); //Nhập vào giá trị cho 3 biến nguyên x, y, z

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

ptr = &x; //Gán địa chỉ của biến x cho con trỏ ptr

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

ptr = &y; //Gán địa chỉ của biến y cho con trỏ ptr

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

ptr = &z; //Gán địa chỉ của biến z cho con trỏ ptr

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

return 0;

}//Tô Thái Linh 20215414

## **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 **&**

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

Mô tả được tạo tự động**

Hình bài 1.2 và kết quả

/\*To Thai Linh 20215414

Ma lop thuc hanh: 732826\*/

#include <stdio.h>

int main() {

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

printf("To Thai Linh 20215414 - K66 DHBK HN\n");

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

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

printf("\t\ta[%d] ", i);//In ra cac xau a[i]

}

printf("\n");

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

printf("\t%p", &a[i]);//In ra dia chi cua 5 phan tu dau

}

printf("\n");

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.

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

Mô tả được tạo tự động

Hình bài 1.3

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

Mô tả được tạo tự động

Hình Kết quả bài 1.3

/\*To Thai Linh 20215414

Ma lop thuc hanh: 732826\*/

#include <stdio.h>

int main() {

int x, y, z;

//Khai bao 3 bien kieu int

int \*ptr;

//Khai bao con tro kieu int

scanf("%d %d %d", &x, &y, &z); //Nhap gia tri cho 3 bien kieu int

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

printf("%d %d %d\n", x, y, z); //In ra gia tri cua 3 bien kieu int

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

\*ptr += 100;//Tang gia tri bien x duoc ptr tro den them 100

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

\*ptr += 100;//Tang gia tri bien y duoc ptr tro den them 100

ptr = &z;//Gan gia tri cua z cho con tro ptr

\*ptr += 100;//Tang gia tri cua bien x duoc ptr tro den them 100

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

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

//In ra gia tri cua 3 bien x, y, z sau khi da tang them 100

return 0;

}

//To Thai Linh 20215414

## **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

Hình bài 1.4

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

Mô tả được tạo tự động

Hình Kết quả bài 1.4

/\*To Thai Linh

Ma lop thuc hanh 732826\*/

#include <stdio.h>

#include<iostream.h>

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

int count = 0;

//Khoi tao mot bien count kieu int co gia tri bang 0

for(int i = 0; i < size; i++) { //Duyet qua tat ca cac phan tu cua mang arr

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

/\*(arr + i) la gia tri cua phan tu thu i trong mang = arr[i]

Neu \*(arr + i) chia het cho 2 thi la so chan \*/

count += 1;

//Tang gia tri cua bien count them 1

}

}

return count; //Tra ve gia tri cua bien count

}

int main() {

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

std::cout<<counteven(arr, 8);

return 0;

}

//To Thai Linh 20215414

## **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

Hình bài 1.5

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

Mô tả được tạo tự động

Hình kết quả bài 1.5

/\*To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

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

double \*max;//Khai bao mot con tro max kieu double

max = a; //Gan đia chi đau cua mang a cho max

if (a==NULL) return NULL; //Neu đia chi cua a NULL thi tra ve NULL

for(int i = 0; i < size; i++) { //Duyet tung phan tu của mang

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

//Neu gia tri cua a[i] > gia tri cua đia chi con tro max tro toi

max = a + i;//Gan đia chi cua gia tri đo cho con tro max

}

}

return max;//Tra lai đia chi con tro max tro toi

}//To Thai Linh 20215414

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].

Hình bài 1.6

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, màn hình, phần mềm

Mô tả được tạo tự động

Hình Kết quả bài 1.6

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

Mô tả được tạo tự động

/\*To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <bits/stdc++.h>

using namespace std;

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

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

for(int i = l; i <= r/2; i++) { //Duyet tat ca các phan tu cua mang

tmp = arr[i]; //Gan giá tri cua arr[i] cho tmp

arr[i] = arr[r - i]; //Gan giá tri cua arr[r - i] cho ar[i]

arr[r - i] = tmp; //Gan giá tri cua tmp cho arr[r - i]

}

}

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

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

for(int i = l; i <= r/2; i++) { //Duyet tat ca các phan tu cua mang

tmp = \*(arr + i); //Gán giá tr? c?a arr[i] cho tmp

\*(arr + i) = \*(arr + r - i);//Gan giá tri cua arr[r - i] cho ar[i]

\*(arr + r - i) = tmp; //Gan giá tri cua tmp cho arr[r - i]

}

}

//To Thai Linh 20215414

int main() {

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

reversearray(arr, 6);

for(int i = 0; i < 6; i++) cout << arr[i] << " ";

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

ptr\_reversearray(arr2, 4);

for(int i = 0; i < 4; i++) cout << 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.

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

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

Hình bài 1.7

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

Mô tả được tạo tự động

Hình Kết quả bài 1.7

/\*To Thai Linh 20215414

Ma lop thuc hanh: 732826\*/

#include <stdio.h>

#include <stdlib.h>

int \*a; //Khai bao mot con tro a kieu int

int n, tmp; //Khai bao 2 bien n va tmp kieu int

int main(){

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

scanf("%d", &n); //Nhap gia tri cua n

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

//Cap phat dong n o nho co kich thuoc sizeof(int) va gan dia chi cho a

for(int i = 0; i < n; i++) //Duyet va nhap tung gia tri cho mang a

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

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

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

printf("%d ", \*(a + i));//In ra gia tri tung phan tu trong mang a

printf("\n");

for(int i = 0; i < n - 1; i++) { //Duyet qua tat ca cac phan tu cua con tro

for(int j = i + 1; j < n; j++) { //Duyet tu phan tu thu i + 1 d?n het

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

/\*So sanh 2 phan tu tu i + 1 cho den het de tim gia tri nho nhat cua

phan tu thu i roi dao gia tri\*/

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)); //In ra gia tri cua tung phan tu sau khi sap xep

printf("\n");

delete [] a; //Giai phong vung nho

return 0;

}

//To Thai Linh 20215414

## **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(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 độngẢnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Hình bài 1.8 và kết quả

/\*To Thai LinhLinh 20215414

Ma lop thuc hanh: 732826\*/

#include <bits/stdc++.h>

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

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

//Cap phat dong m con tro kieu int

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

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

//Cap phat dong n ô nho duoc tro boi m[i]

}

}

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);

//Nhap du lieu cho mt[i][j] co dia chi \*(m + 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));//In du lieu mang 2 chieu

}

printf("\n");

}

}

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

int tong = 0;

//Khoi tao mot bien tong kieu int bang 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);

}

}

//Kiem tra dieu kien neu mt[i][j] chan thi cong vao tong

}

return tong;

}

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

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

delete [] \*(mt + i); //Xóa tung hang cua mt

}

delete [] mt;//Xoa mang luu dia chi cua tung hang

}

int main(){

int m, n, \*\*mt;//Khai bao 2 bien kieu int và 1 con tro kieu int

printf("Enter m, n = ");

scanf("%d%d", &m, &n);//Nhap gia tri cua m và n

allocate\_mem(&mt, m, n);//Goi den ham cap phat bo nho cho mang 2 chieu

input(mt, m, n);//Goi ham nhap mang

output(mt, m, n);//Goi ham in ra mang

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

free\_mem(mt, m, n);//Giai phong bo nho

return 0;

}

//To Thai Linh 20215414

## **Bài tập 1.9**: 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

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

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình bài 1.9

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

Mô tả được tạo tự động

Hình bài 1.9 test case 1

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Hình bài 1.9 test case 2

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

Mô tả được tạo tự động

Hình Bài 1.9 test case 3

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

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, văn bản, thiết kế

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình bài 1.9 test case 4

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

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, màu đen, thiết kế

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, thiết kế, thuật in máy

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, văn bản, thiết kế

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, màu đen, thiết kế

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, màu đen, thiết kế, tác phẩm nghệ thuật

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, thiết kế

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, văn bản, thiết kế

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Hình bài 1.9 test case 5

/\*To Thai Linh 20215414

Ma lop thuc hanh: 732826

Bai 1.9\*/

#include <bits/stdc++.h>

void printSolution(int \*arr, int size) {

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

//Duyet tat ca cac phan tu cua day

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

//Duyet cac day con cua day

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

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

//In ra day con tu i den j

}

printf("\n");

}

}

}

int main() {

printf("To Thai Linh 2021514 - K66 DHBKHN\n");

int n;//Khai bao mot bien n kieu int

int \*arr;//Khai bao mot con tro arr kieu int

scanf("%d", &n);//Nhap gia tri cua n

//Cap phat dong bo nho n phan tu kieu int

arr = new int[n];

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

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

//Nhap gia tri cua tung phan tu

}

printf("\nKet qua la:\n");

printSolution(arr, n);//Goi ham in ra ket qua

delete [] arr;//Giai phong vung nho duoc cap phat

return 0;

}

//To Thai Linh 20215414

## **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, Phông chữ, màn hình

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

Hình bài 1.10

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

Mô tả được tạo tự động

Hình bài 1.10 test case 1

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Hình bài 1.10 test case 2

Ảnh có chứa ảnh chụp màn hình, văn bản, màu đen

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Hình bài 1.10 test case 3

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

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, thiết kế, thuật in máy

Mô tả được tạo tự động

Hình bài 1.10 test case 4

/\*To Thai Linh 20215414

Ma lop thuc hanh: 732826

Bai 1.10\*/

#include <bits/stdc++.h>

using namespace std;

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

//Ham cap phat bo nho dong cho mang 2 chieu

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

//Cap phat bo nho cho cac hang

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

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

//cap phat bo nho cho cac cot

}

}

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

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

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

cin>>mt[i][j];

//Nhap gia tri cua tung phan tu

}

}

}

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

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

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

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

//in ra gia tri tung phan tu

}

cout<<endl;

}

}

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

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

delete [] \*(mt + i);

//giai phong bo nho cua cac hang

}

delete [] mt;//giai phong bo nho cac cot

}

int main() {

cout<<"To Thai Linh 20215414 - K66 DHBKHN"<<endl;

int n;//khai bao mot bien n kieu int

cin >> n;//nhap gia tri cua n

int \*\*matrix1, \*\*matrix2, \*\*sumMatrix, \*\*productMatrix;

//khai bao 4 bien con tro cap 2 kieu int

allocate(&matrix1, n);//cap phat bo nho cho matrix1

allocate(&matrix2, n);//cap phat bo nho cho matrix2

allocate(&sumMatrix, n);//cap phat bo nho cho sumMatrix

allocate(&productMatrix, n);//cap phat bo nho cho productMatrix

input(matrix1, n);//nhap gia tri cua matrix1

input(matrix2, n);//nhap gia tri cua matrix2

//Tinh tong 2 ma tran

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

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

sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

//Tinh tich 2 ma tran

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

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

productMatrix[i][j] = 0;

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

productMatrix[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

cout<<"Ket qua la:"<<endl;

output(sumMatrix, n);//in ra ma tran tong

output(productMatrix, n);//in ra ma tran tich

free\_mem(matrix1, n);//giai phong ma tran 1

free\_mem(matrix2, n);//giaiphong ma tran 2

free\_mem(sumMatrix, n);//giai phong ma tran tong

free\_mem(productMatrix, n);//giai phong ma tran tich

return 0;

}//To Thai Linh 20215414