

Contents

[Bài thực hành số 1 – Tuần 31 3](#_Toc23478)

[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. 3](#_Toc8227)

[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](#_Toc2142)

[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](#_Toc7800)

[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](#_Toc10412)

[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. 10](#_Toc14695)

[Bài tập 1.6. Viết hàm đảo ngược một mảng 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]. 13](#_Toc24803)

[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 các mảng trước và sau khi sắp xếp. Yêu cầu chỉ sử dụng con trỏ để truy cập mảng, không truy cập theo index mảng. 16](#_Toc26199)

[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](#_Toc1908)

[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: 27](#_Toc24608)

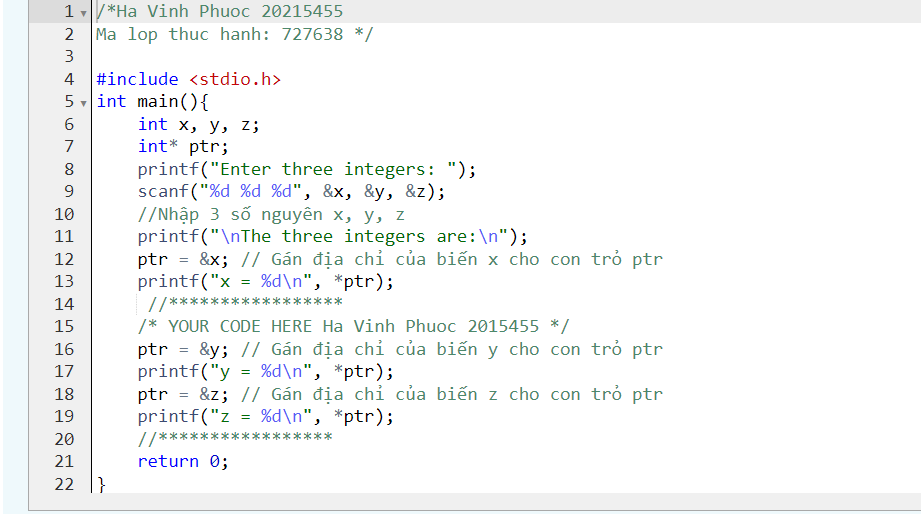
[Bài tập 1.10. Viết chương trình nhập vào 2 ma trận vuông cùng kích thức n\*n, trong đó n nhập từ bàn phím. Sau đó tính tổng và tích của hai ma trận đó và đưa kết quả ra màn hình. 30](#_Toc28866)

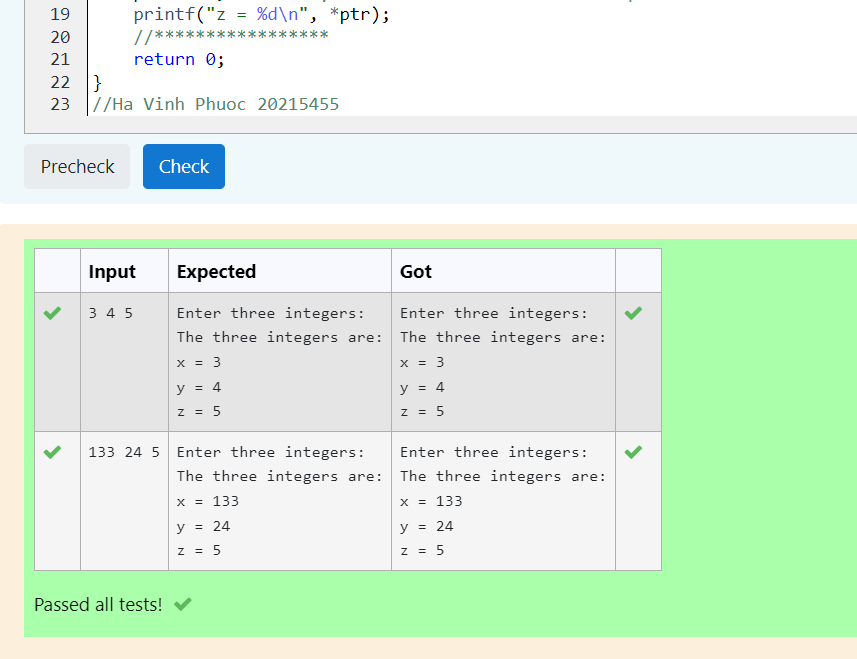
# Bài thực hành số 1 – Tuần 31

## 

## **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.





/\*Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <stdio.h>

int main(){

int x, y, z;

int\* ptr;

printf("Enter three integers: ");

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

//Nhập 3 số 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);

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

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;

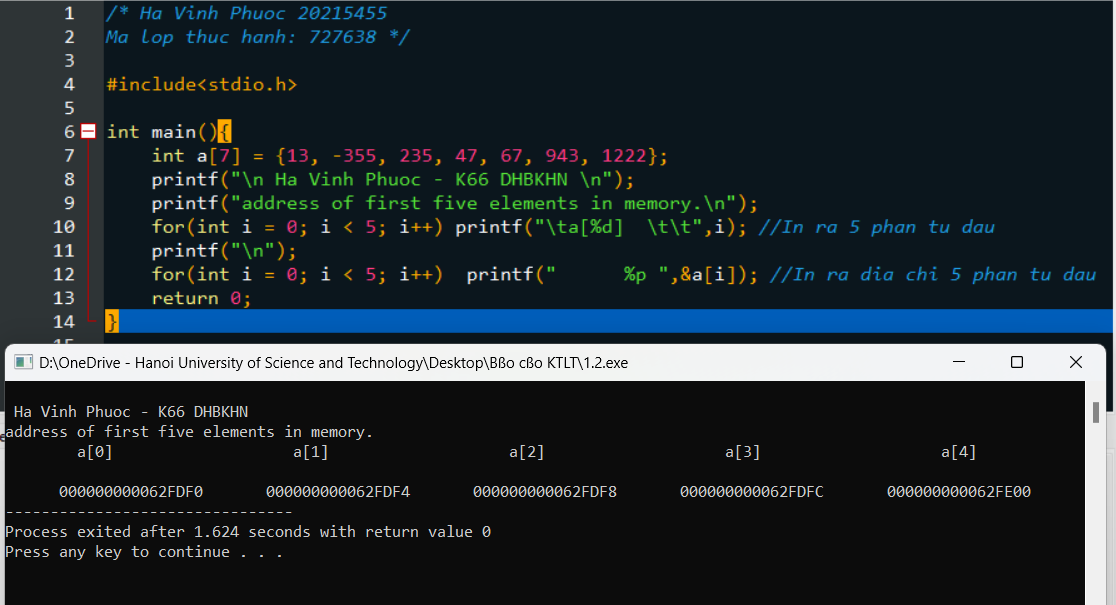
}

## 

## **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 &



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include<stdio.h>

int main(){

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

printf("\n Ha Vinh Phuoc - K66 DHBKHN \n");

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

for(int i = 0; i < 5; i++) printf("\ta[%d] \t\t",i); //In ra 5 phan tu dau

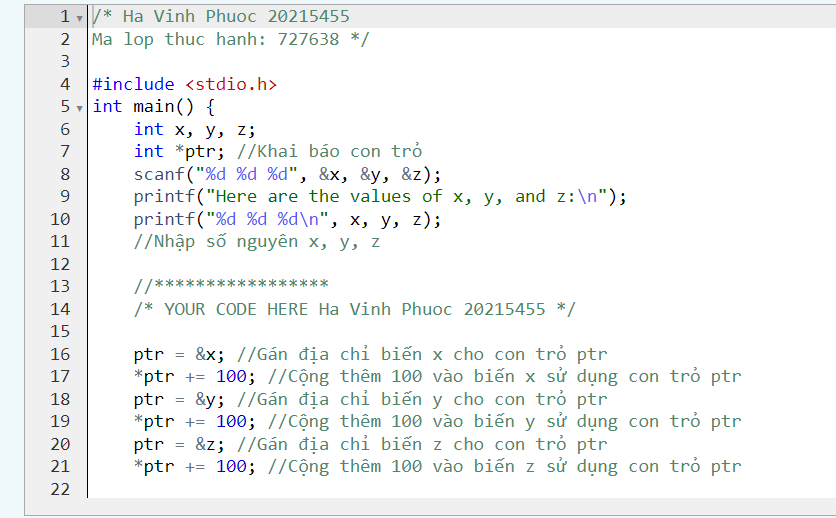
printf("\n");

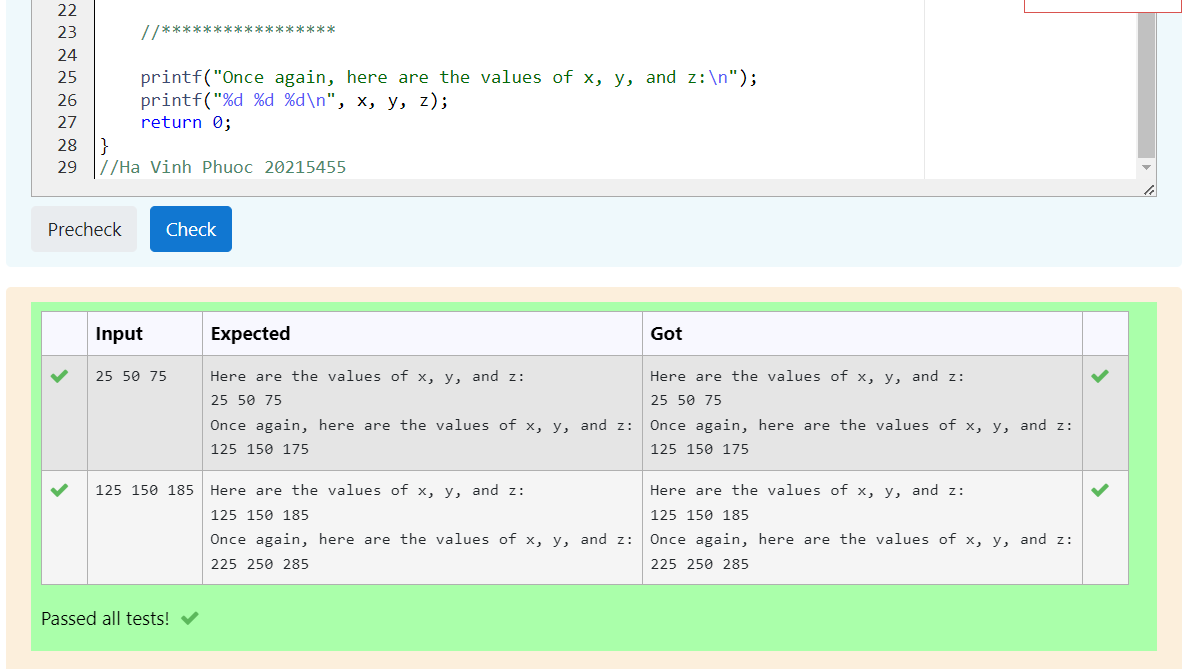
for(int i = 0; i < 5; i++) printf(" %p ",&a[i]); //In ra dia chi 5 phan tu dau

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.





/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <stdio.h>

int main() {

int x, y, z;

int \*ptr; //Khai báo con trỏ

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);

//Nhập số nguyên x, y, z

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

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

\*ptr += 100; //Cộng thêm 100 vào biến x sử dụng con trỏ ptr

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

\*ptr += 100; //Cộng thêm 100 vào biến y sử dụng con trỏ ptr

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

\*ptr += 100; //Cộng thêm 100 vào biến z sử dụng con trỏ ptr

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

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

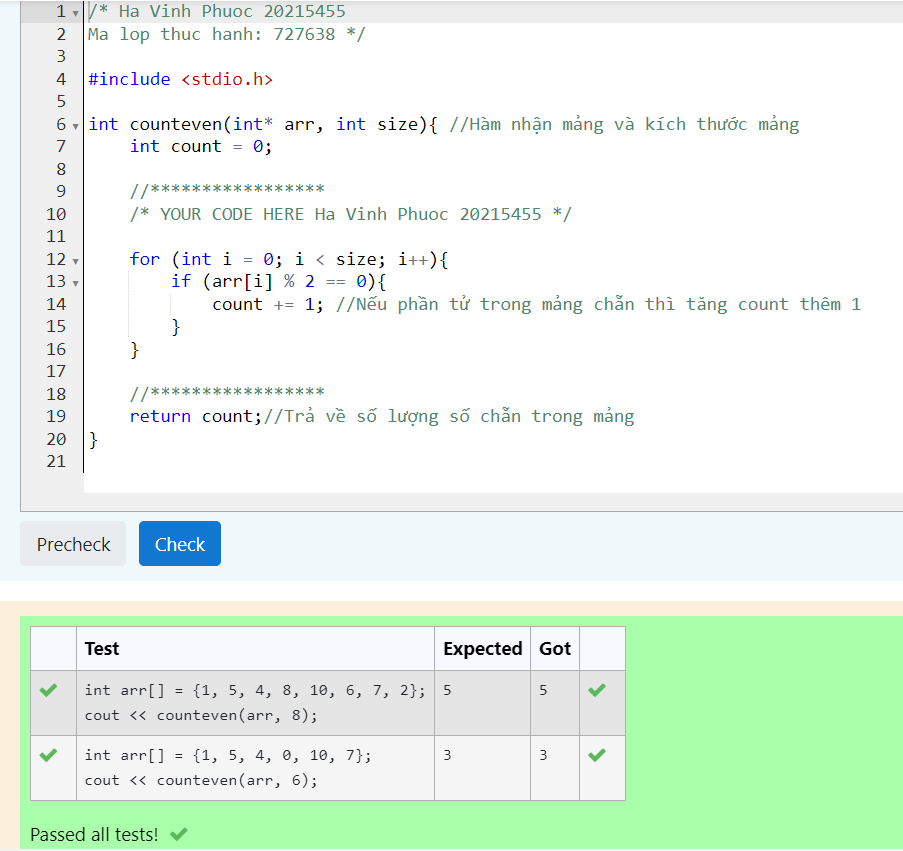
printf("%d %d %d\n", x, y, z);

return 0;

}

//Ha Vinh Phuoc 20215455

## **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.



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <stdio.h>

int counteven(int\* arr, int size){ //Hàm nhận mảng và kích thước mảng

int count = 0;

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

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

if (arr[i] % 2 == 0){

count += 1; //Nếu phần tử trong mảng chẵn thì tăng count thêm 1

}

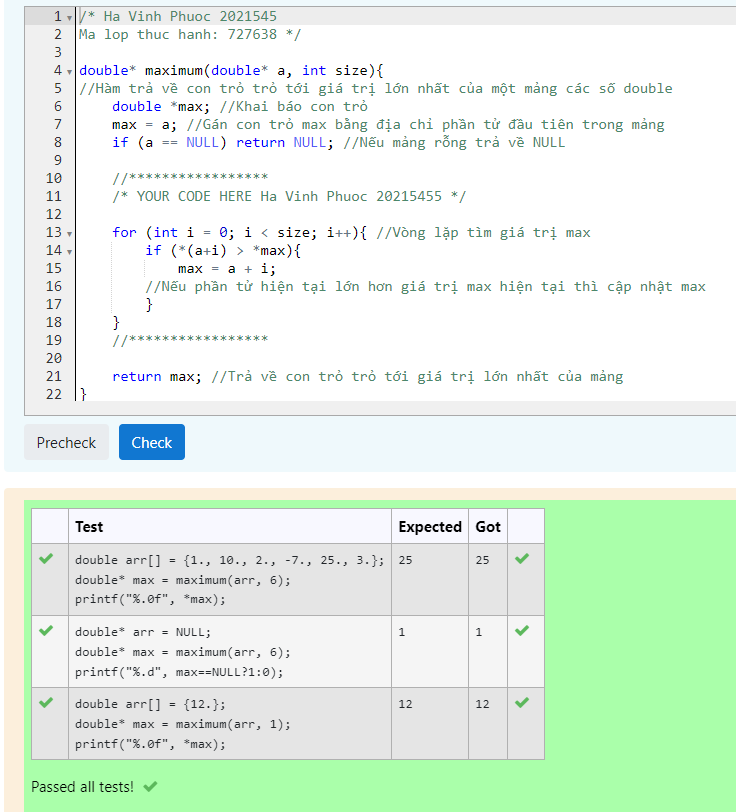
}

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

return count; //Trả về số lượng số chẵn trong mảng

}

## **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.



//\* Ha Vinh Phuoc 2021545

Ma lop thuc hanh: 727638 \*/

#include <stdio.h>

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

//Hàm trả về con trỏ trỏ tới giá trị lớn nhất của một mảng các số double

double \*max; //Khai báo con trỏ

max = a; //Gán con trỏ max bằng địa chỉ phần tử đầu tiên trong mảng

if (a == NULL) return NULL; //Nếu mảng rỗng trả về NULL

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

for (int i = 0; i < size; i++){ //Vòng lặp tìm giá trị max

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

max = a + i;

//Nếu phần tử hiện tại lớn hơn giá trị max hiện tại thì cập nhật max

}

}

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

return max; //Trả về con trỏ trỏ tới giá trị lớn nhất của mảng

}

int main(){

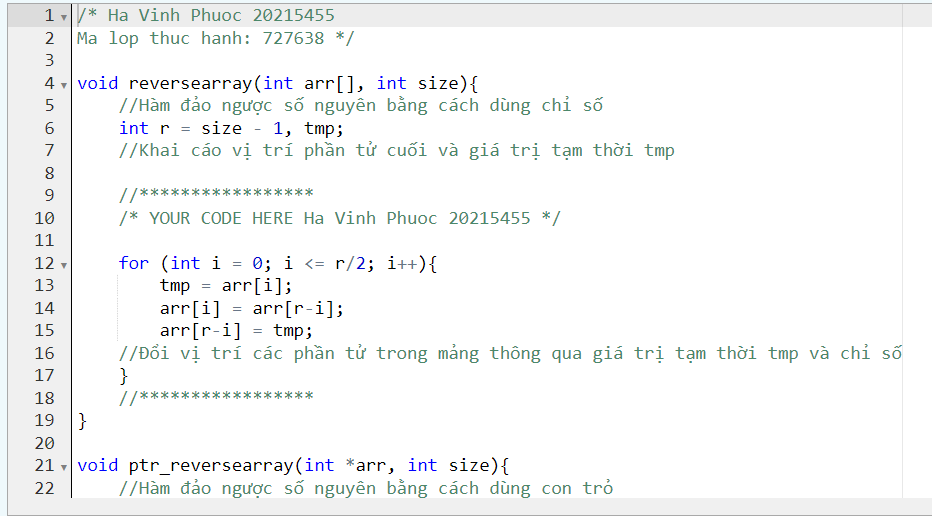
double arr[] = {1., 10., 2., -7., 25., 3.};

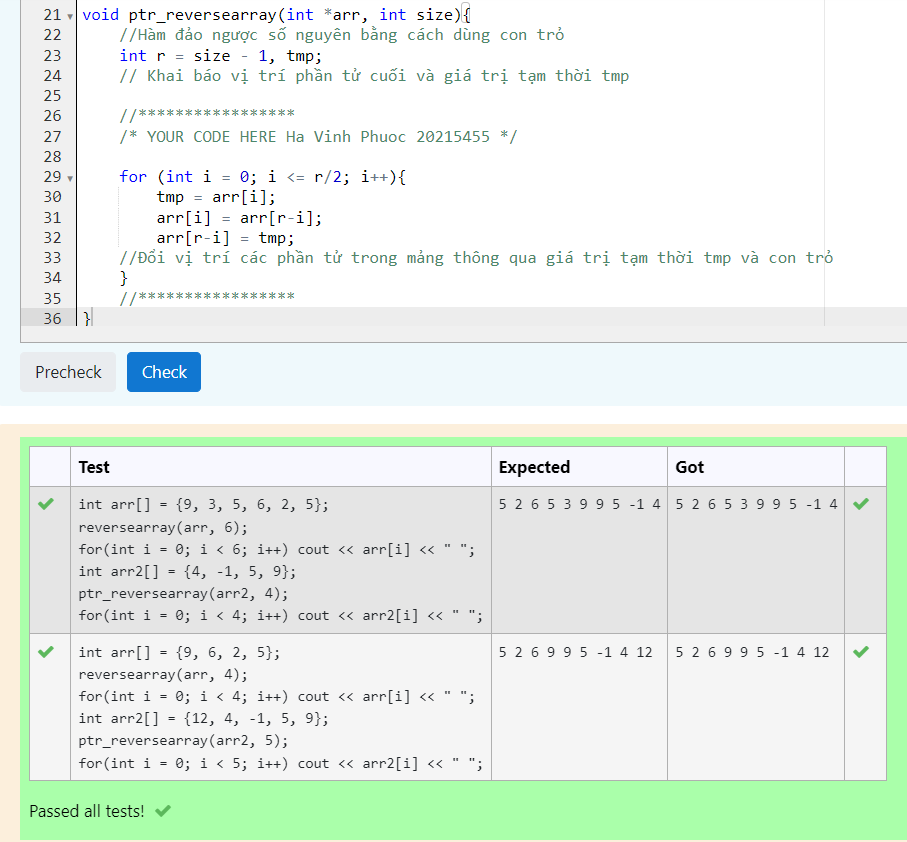
double\* max = maximum(arr, 6);

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

}

## **Bài tập 1.6.**Viết hàm đảo ngược một mảng 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].





/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <stdio.h>

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

//Hàm đảo ngược số nguyên bằng cách dùng chỉ số

int r = size - 1, tmp;

//Khai cáo vị trí phần tử cuối và giá trị tạm thời tmp

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

for (int i = 0; i <= r/2; i++){

tmp = arr[i];

arr[i] = arr[r-i];

arr[r-i] = tmp;

//Đổi vị trí các phần tử trong mảng thông qua giá trị tạm thời tmp và chỉ số

}

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

}

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

//Hàm đảo ngược số nguyên bằng cách dùng con trỏ

int r = size - 1, tmp;

// Khai báo vị trí phần tử cuối và giá trị tạm thời tmp

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

for (int i = 0; i <= r/2; i++){

tmp = arr[i];

arr[i] = arr[r-i];

arr[r-i] = tmp;

//Đổi vị trí các phần tử trong mảng thông qua giá trị tạm thời tmp và con trỏ

}

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

}

int main(){

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

reversearray(arr, 6);

for(int i = 0; i < 6; i++) printf("%d ", arr[i]);

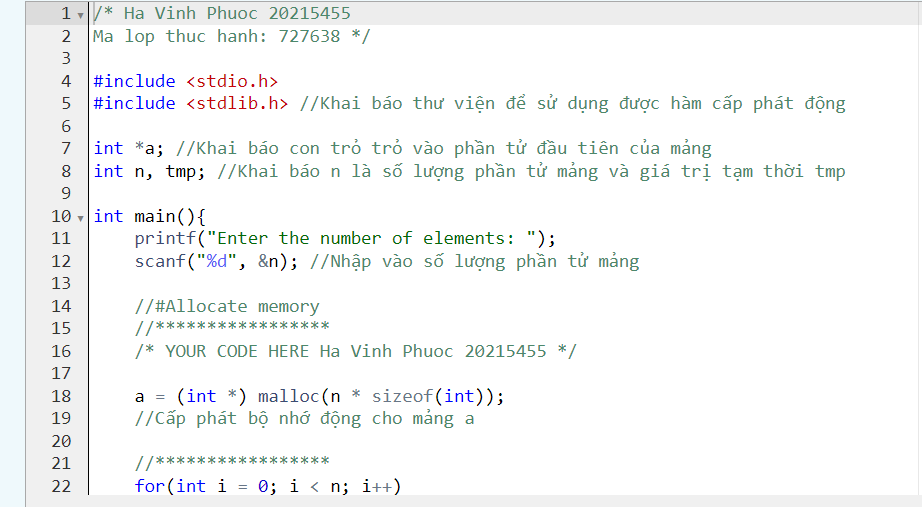
int arr2[] = {4, -1, 5, 9};

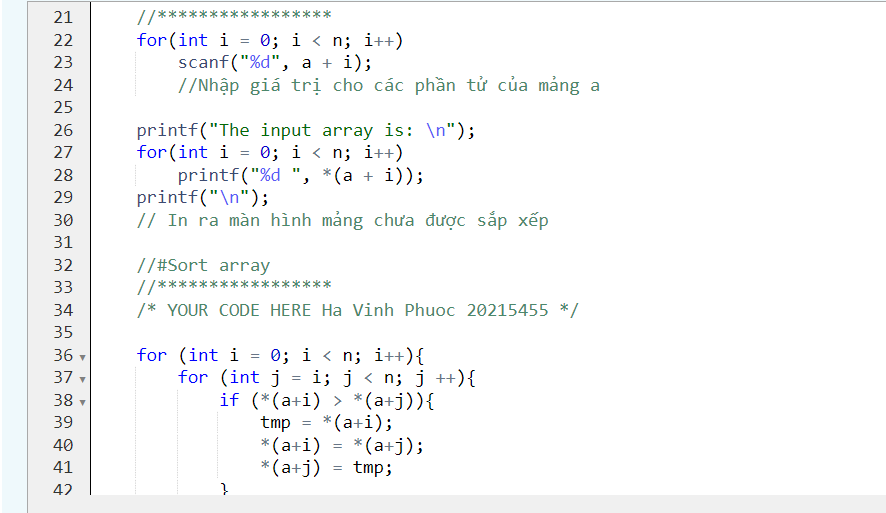
ptr\_reversearray(arr2, 4);

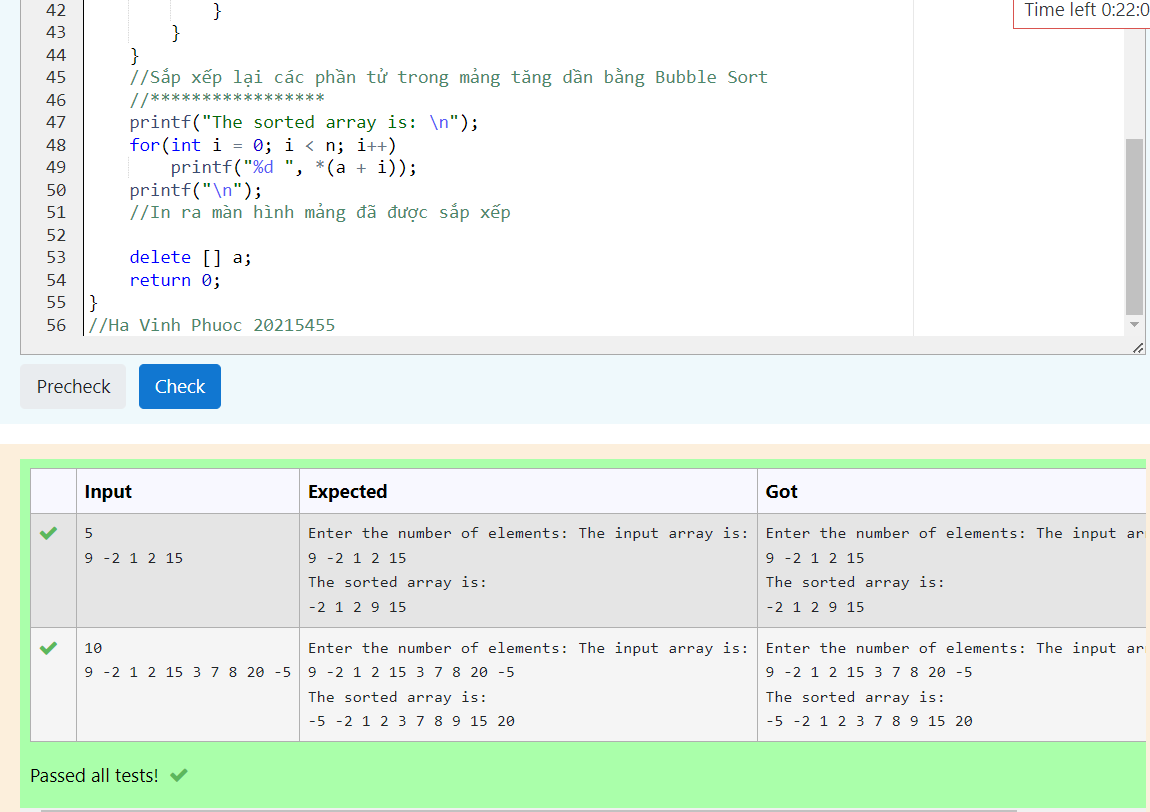
for(int i = 0; i < 4; i++) printf("%d ", arr2[i]);

}

## **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 các mảng trước và sau khi sắp xếp. Yêu cầu chỉ sử dụng con trỏ để truy cập mảng, không truy cập theo index mảng.







/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <stdio.h>

#include <stdlib.h> //Khai báo thư viện để sử dụng được hàm cấp phát động

int \*a; //Khai báo con trỏ trỏ vào phần tử đầu tiên của mảng

int n, tmp; //Khai báo n là số lượng phần tử mảng và giá trị tạm thời tmp

int main(){

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

scanf("%d", &n); //Nhập vào số lượng phần tử mảng

//#Allocate memory

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

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

//Cấp phát bộ nhớ động cho mảng a

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

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

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

//Nhập giá trị cho các phần tử của mảng a

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

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

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

printf("\n");

// In ra màn hình mảng chưa được sắp xếp

//#Sort array

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

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

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

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

tmp = \*(a+i);

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

\*(a+j) = tmp;

}

}

}

//Sắp xếp lại các phần tử trong mảng tăng dần bằng Bubble Sort

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

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

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

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

printf("\n");

//In ra màn hình mảng đã được sắp xếp

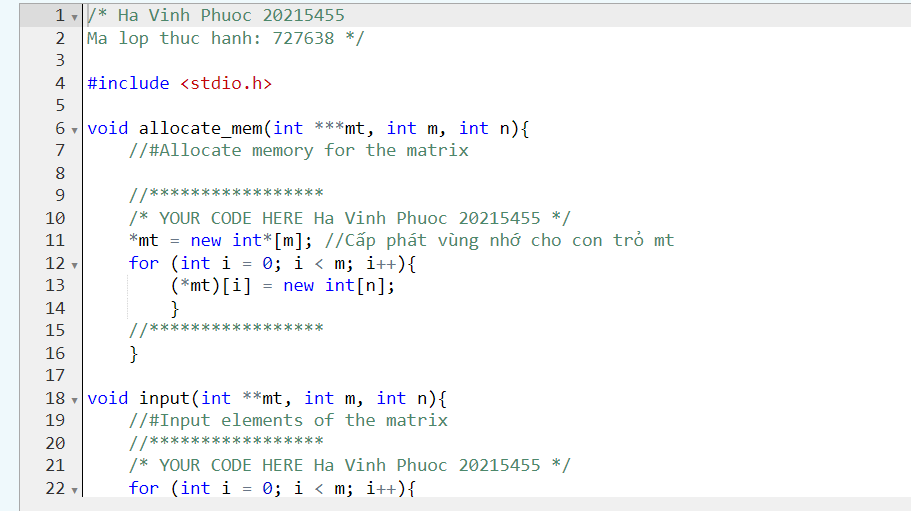
delete [] a;

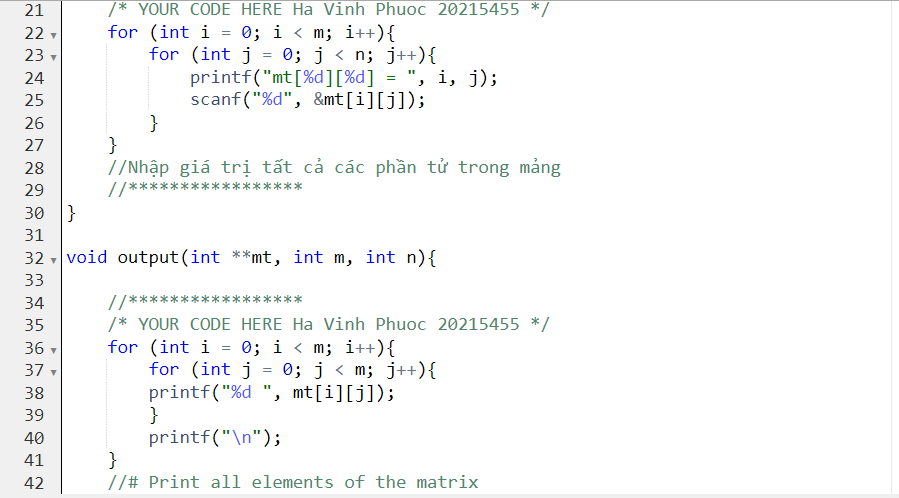
return 0;

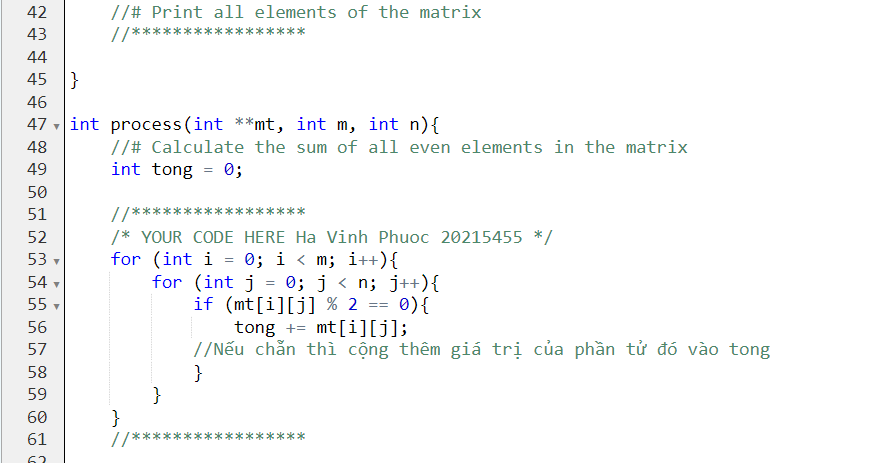
}

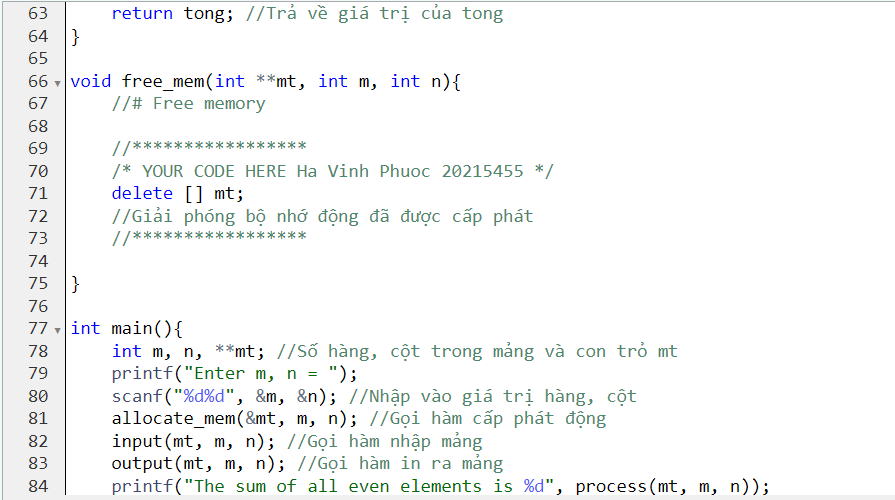
//Ha Vinh Phuoc 20215455

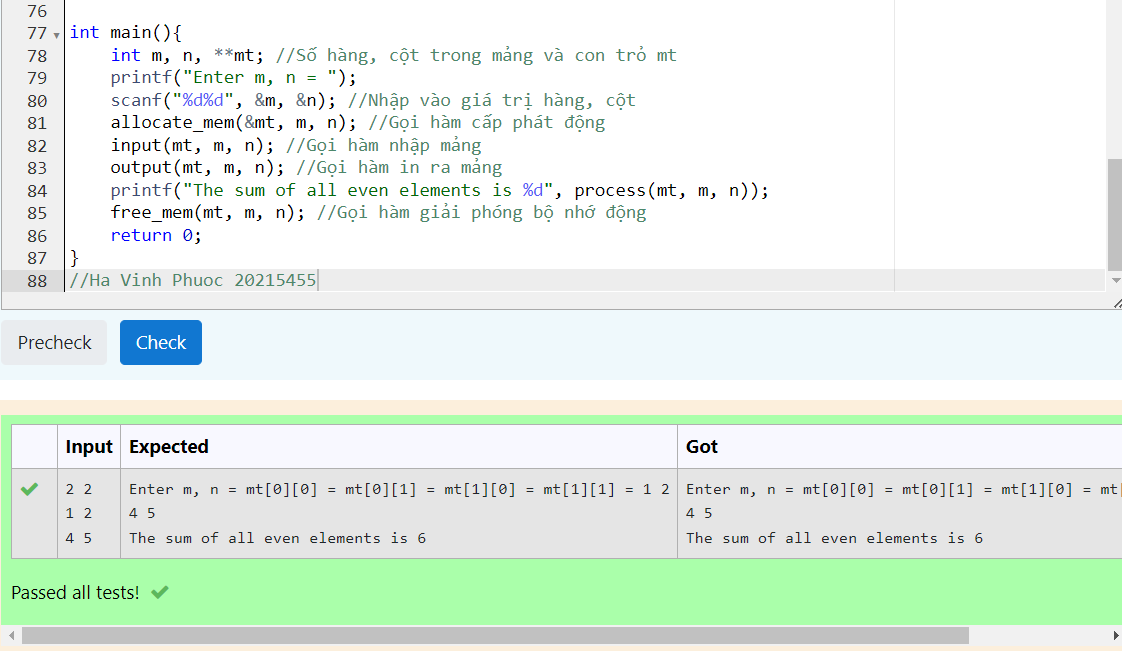
## **Bài tập 1.8.**Viết chương trình nhập vào một ma trận 2 chiều kích thước m\*n với m và n nhập từ bàn phím. Sau đó đưa ra tổng các phần tử chẵn của ma trận đó. Lưu ý: Khi viết hàm cấp phát bộ nhớ cho một ma trận hai chiều biểu diễn bởi con trỏ int \*\*mt, nếu ta truyền con trỏ theo kiểu địa chỉ void 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.











/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <stdio.h>

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

//#Allocate memory for the matrix

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

\*mt = new int\*[m]; //Cấp phát vùng nhớ cho con trỏ mt

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

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

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]);

}

}

//Nhập giá trị tất cả các phần tử trong mảng

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

}

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

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

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

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

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

}

printf("\n");

}

//# Print all elements of the matrix

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

}

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

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

int tong = 0;

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

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

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

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

tong += mt[i][j];

//Nếu chẵn thì cộng thêm giá trị của phần tử đó vào tong

}

}

}

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

return tong; //Trả về giá trị của tong

}

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

//# Free memory

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

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

delete [] mt;

//Giải phóng bộ nhớ động đã được cấp phát

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

}

int main(){

int m, n, \*\*mt; //Số hàng, cột trong mảng và con trỏ mt

printf("Enter m, n = ");

scanf("%d%d", &m, &n); //Nhập vào giá trị hàng, cột

allocate\_mem(&mt, m, n); //Gọi hàm cấp phát động

input(mt, m, n); //Gọi hàm nhập mảng

output(mt, m, n); //Gọi hàm in ra mảng

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

free\_mem(mt, m, n); //Gọi hàm giải phóng bộ nhớ động

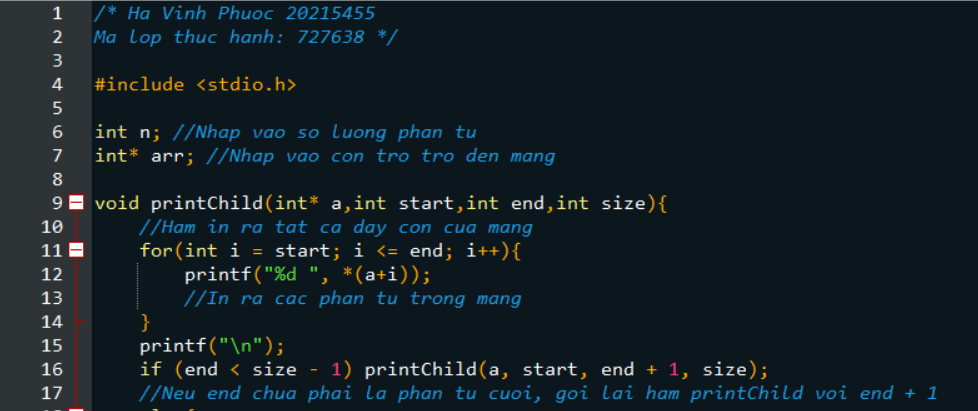
return 0;

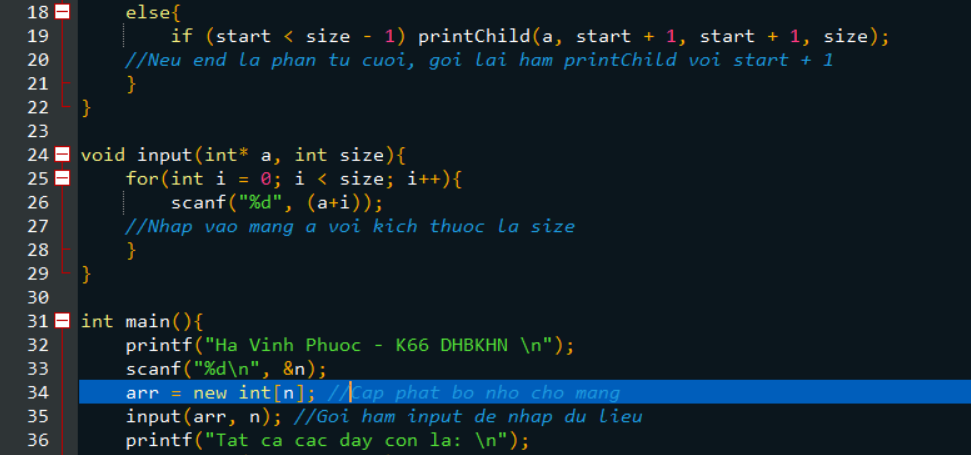
}

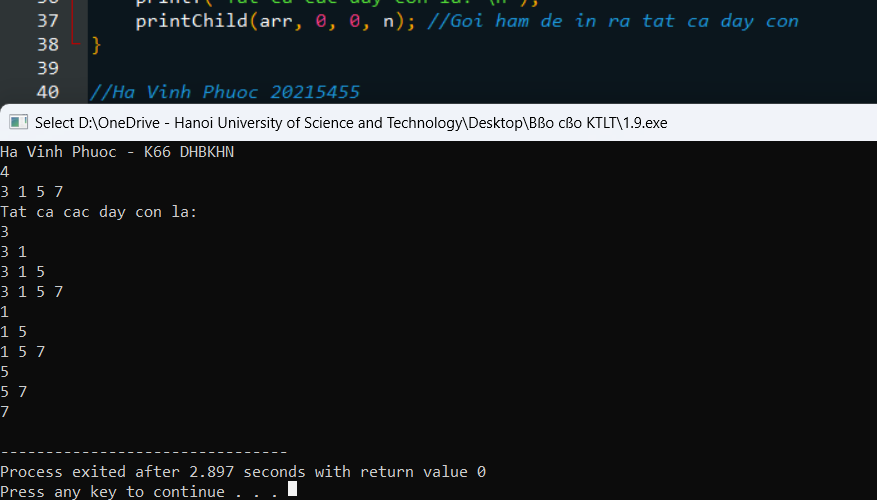
//Ha Vinh Phuoc 20215455

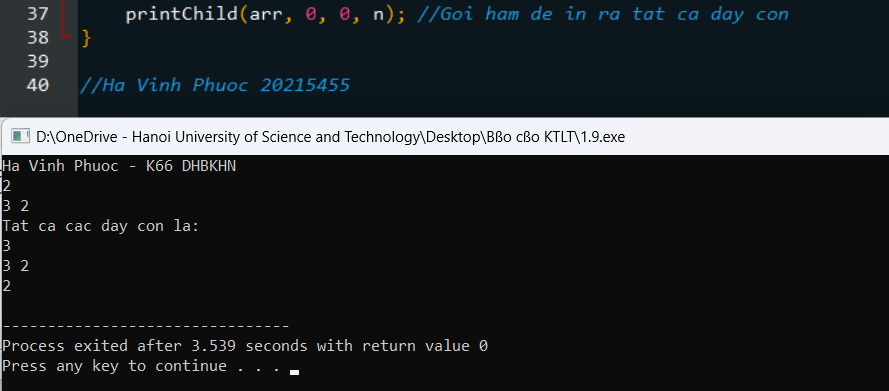
## **Bài tập 1.9.**Viết chương trình in ra tất cả các dãy con của một dãy cho trước. Ví dụ dãy 1 3 4 2 có các dãy con sau:

1  
1 3  
1 3 4  
1 3 4 2  
3  
3 4  
3 4 2  
4  
4 2  
2









//\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <stdio.h>

int n; //Nhap vao so luong phan tu

int\* arr; //Nhap vao con tro tro den mang

void printChild(int\* a,int start,int end,int size){

//Ham in ra tat ca day con cua mang

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

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

//In ra cac phan tu trong mang

}

printf("\n");

if (end < size - 1) printChild(a, start, end + 1, size);

//Neu end chua phai la phan tu cuoi, goi lai ham printChild voi end + 1

else{

if (start < size - 1) printChild(a, start + 1, start + 1, size);

//Neu end la phan tu cuoi, goi lai ham printChild voi start + 1

}

}

void input(int\* a, int size){

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

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

//Nhap vao mang a voi kich thuoc la size

}

}

int main(){

printf("Ha Vinh Phuoc - K66 DHBKHN");

scanf("%d", &n);

arr = new int[n]; //Cap phat bo nho cho mang

printf(“Tat ca cac day con la: \n”);

input(arr, n); //Goi ham input de nhap du lieu

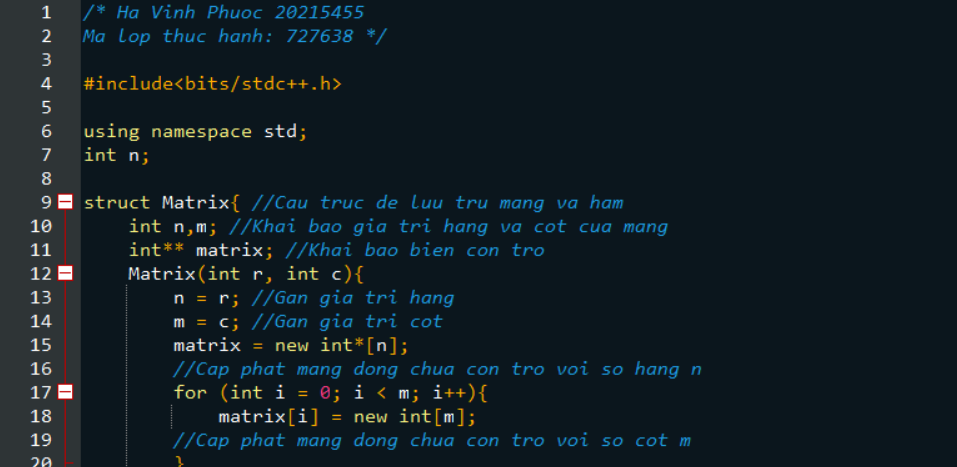
printChild(arr, 0, 0, n); //Goi ham de in ra tat ca day con

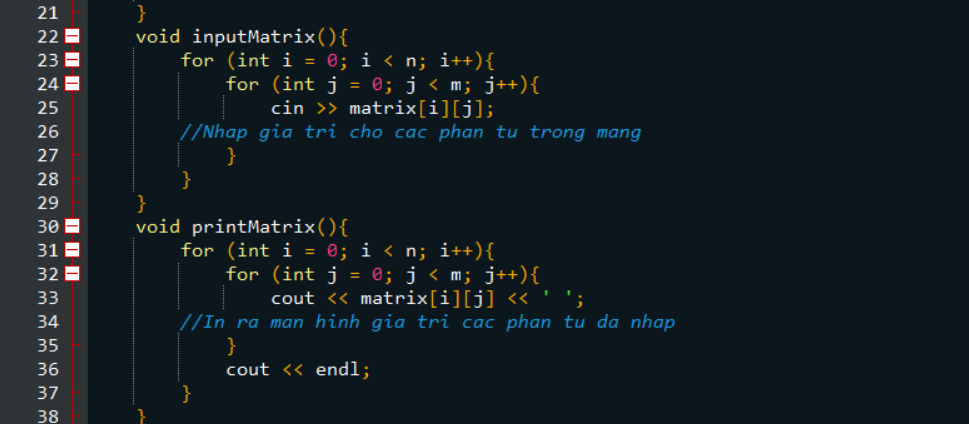
}

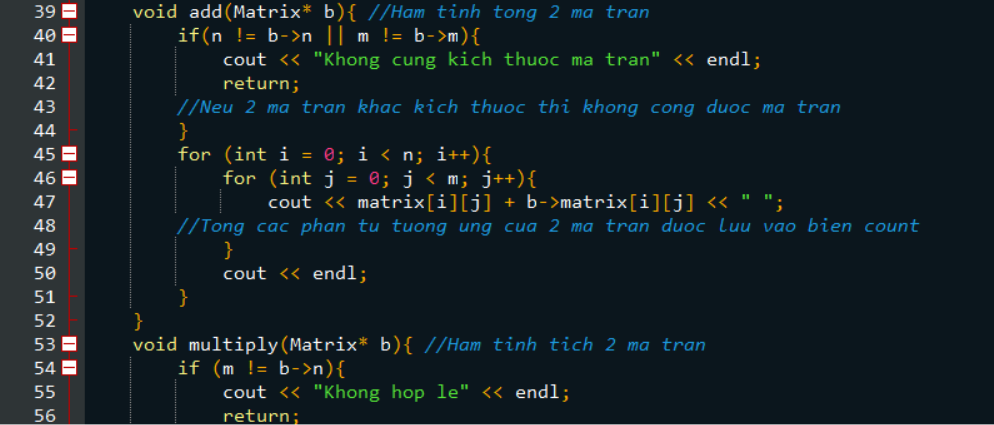
//Ha Vinh Phuoc 20215455

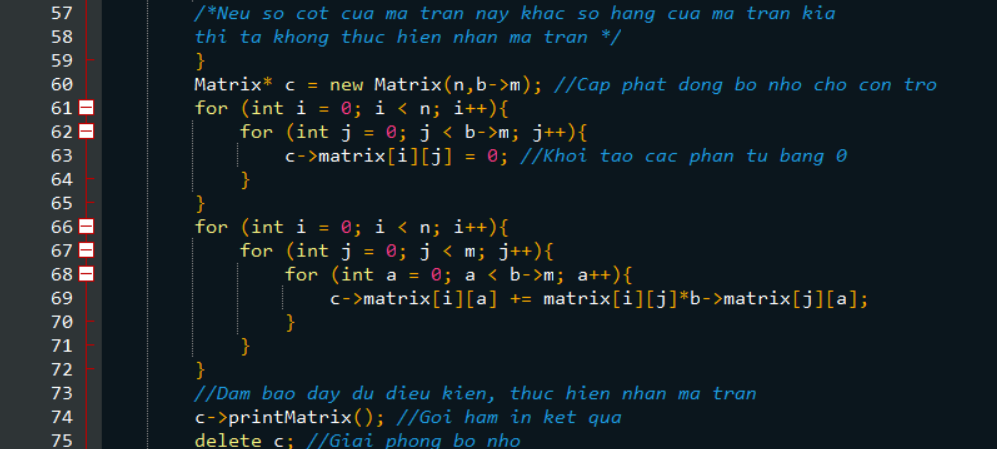
## **Bài tập 1.10.**Viết chương trình nhập vào 2 ma trận vuông cùng kích thức n\*n, trong đó n nhập từ bàn phím. Sau đó tính tổng và tích của hai ma trận đó và đưa kết quả ra màn hình.

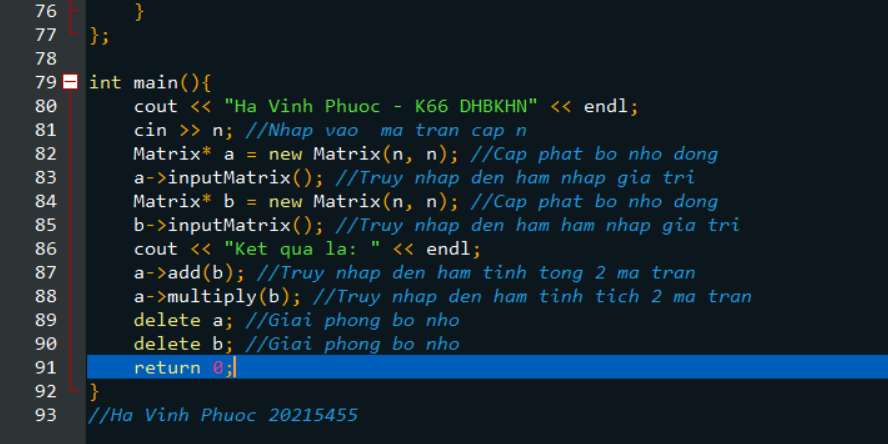
## Yêu cầu sử dụng cấp phát động để cấp phát bộ nhớ cho các ma trận.

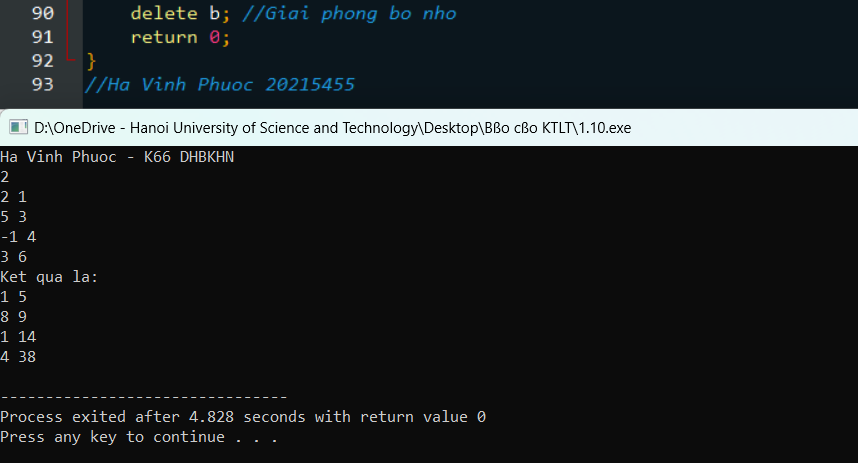


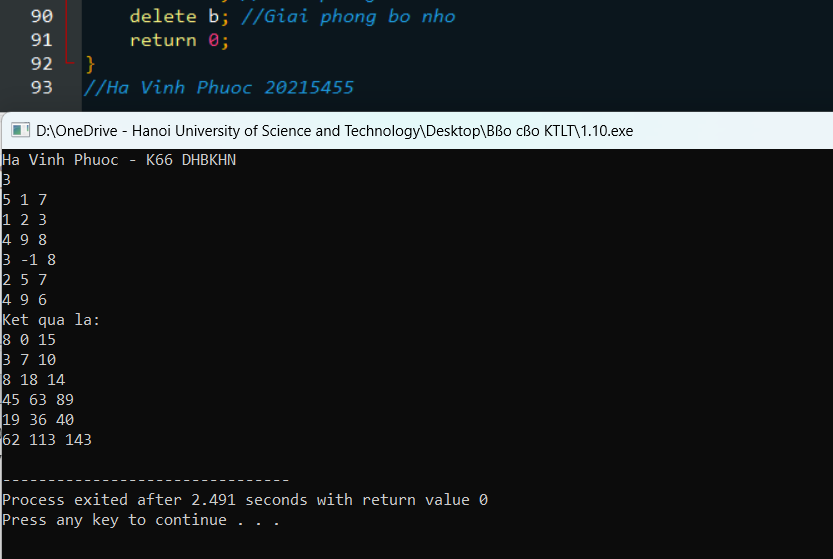












/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include<bits/stdc++.h>

using namespace std;

int n;

struct Matrix{ //Cau truc de luu tru mang va ham

int n,m; //Khai bao gia tri hang va cot cua mang

int\*\* matrix; //Khai bao bien con tro

Matrix(int r, int c){

n = r; //Gan gia tri hang

m = c; //Gan gia tri cot

matrix = new int\*[n];

//Cap phat mang dong chua con tro voi so hang n

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

matrix[i] = new int[m];

//Cap phat mang dong chua con tro voi so cot m

}

}

void inputMatrix(){

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

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

cin >> matrix[i][j];

//Nhap gia tri cho cac phan tu trong mang

}

}

}

void printMatrix(){

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

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

cout << matrix[i][j] << ' ';

//In ra man hinh gia tri cac phan tu da nhap

}

cout << endl;

}

}

void add(Matrix\* b){ //Ham tinh tong 2 ma tran

if(n != b->n || m != b->m){

cout << "Khong cung kich thuoc ma tran" << endl;

return;

//Neu 2 ma tran khac kich thuoc thi khong cong duoc ma tran

}

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

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

cout << matrix[i][j] + b->matrix[i][j] << " ";

//Tong cac phan tu tuong ung cua 2 ma tran duoc luu vao bien count

}

cout << endl;

}

}

void multiply(Matrix\* b){ //Ham tinh tich 2 ma tran

if (m != b->n){

cout << "Khong hop le" << endl;

return;

/\*Neu so cot cua ma tran nay khac so hang cua ma tran kia

thi ta khong thuc hien nhan ma tran \*/

}

Matrix\* c = new Matrix(n,b->m); //Cap phat dong bo nho cho con tro

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

for (int j = 0; j < b->m; j++){

c->matrix[i][j] = 0; //Khoi tao cac phan tu bang 0

}

}

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

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

for (int a = 0; a < b->m; a++){

c->matrix[i][a] += matrix[i][j]\*b->matrix[j][a];

}

}

}

//Dam bao day du dieu kien, thuc hien nhan ma tran

c->printMatrix(); //Goi ham in ket qua

delete c; //Giai phong bo nho

}

};

int main(){

cout << "Ha Vinh Phuoc - K66 DHBKHN" << endl;

cin >> n; //Nhap vao ma tran cap n

Matrix\* a = new Matrix(n, n); //Cap phat bo nho dong

a->inputMatrix(); //Truy nhap den ham nhap gia tri

Matrix\* b = new Matrix(n, n); //Cap phat bo nho dong

b->inputMatrix(); //Truy nhap den ham ham nhap gia tri

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

a->add(b); //Truy nhap den ham tinh tong 2 ma tran

a->multiply(b); //Truy nhap den ham tinh tich 2 ma tran

delete a; //Giai phong bo nho

delete b; //Giai phong bo nho

return 0;

}

//Ha Vinh Phuoc 20215455