Mục lục nội dung

[Bài thực hành số XX – Tuần XX 3](#_Toc161238431)

[BÀI TẬP TRÊN LAP 3](#_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. 3](#_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. 7](#_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. 7](#_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. 7](#_Toc161238437)

[**Bài tập 1.6.** Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ. 7](#_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. 8](#_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. 8](#_Toc161238440)

[**BÀI TẬP VỀ NHÀ** 8](#_Toc161238441)

[**Bài tập 1.9.** Viết chương trình in ra tất cả các dãy con của một dãy cho trước. 8](#_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. 8](#_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 7](#_Toc161238454)

[Hình 4 Bài 1.4 Hàm trả về số phần tử chẵn trong mảng 7](#_Toc161238455)

[Hình 5 Bài 1.5 Dùng con trỏ tìm phần tử lớn nhất trong mảng 7](#_Toc161238456)

[Hình 6 Bài 1.6 Hàm đảo ngược các phần tử trong mảng dùng chỉ số và con trỏ 7](#_Toc161238457)

[Hình 7 Bài 1.7 Dùng con trỏ sắp xếp mảng theo thứ tự tăng dần 8](#_Toc161238458)

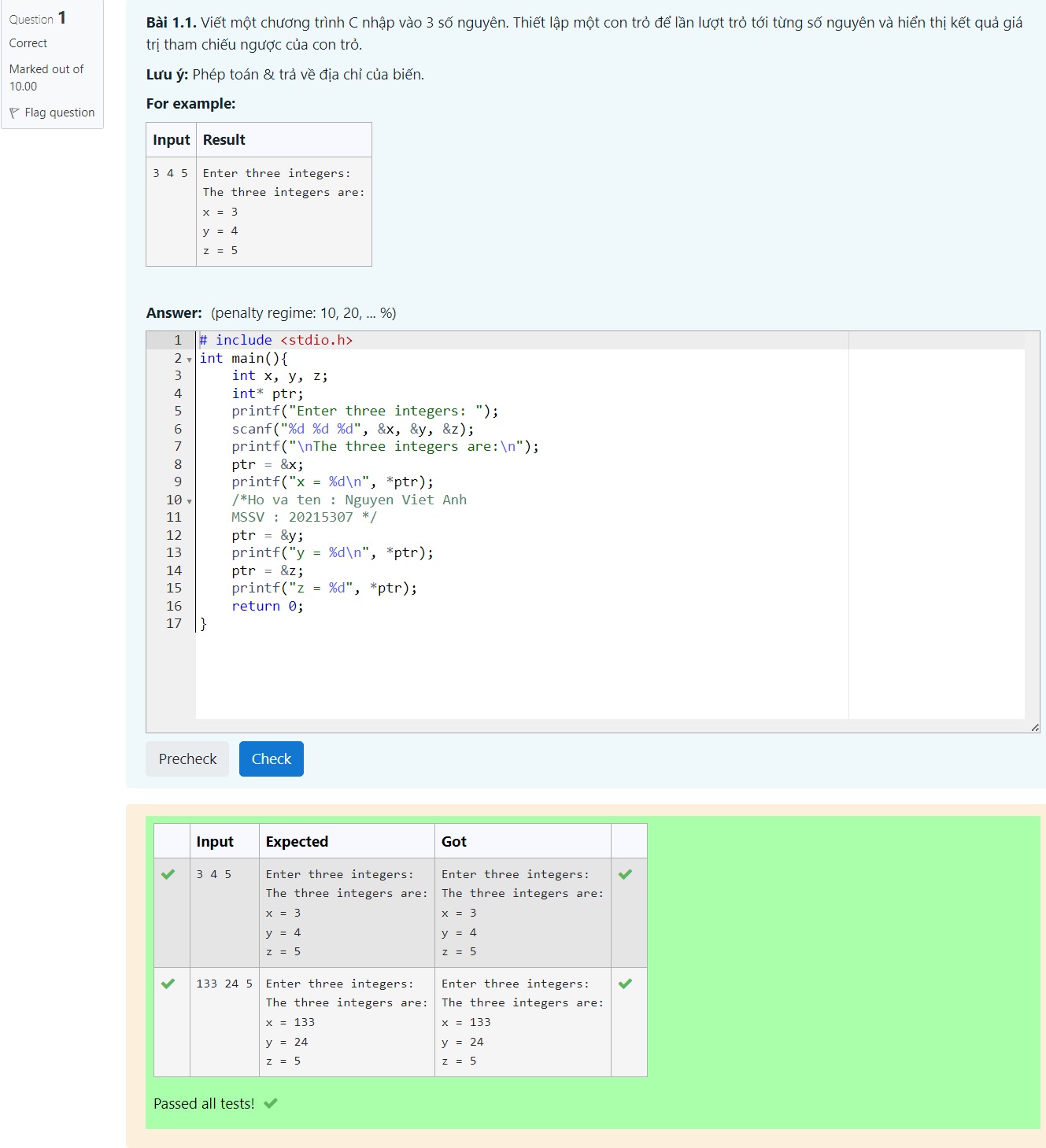
[Hình 8 Bài 1.8 Nhập mảng 2 chiều rồi tính tổng các phần tử chẵn 8](#_Toc161238459)

# Bài thực hành số 1 – Tuần 9

# 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**.



Hình Bài 1.1 In ra 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);

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

ptr = &x;

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

/\*Ho va ten : Nguyen Viet Anh

MSSV : 20215307 \*/

ptr = &y;

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

ptr = &z;

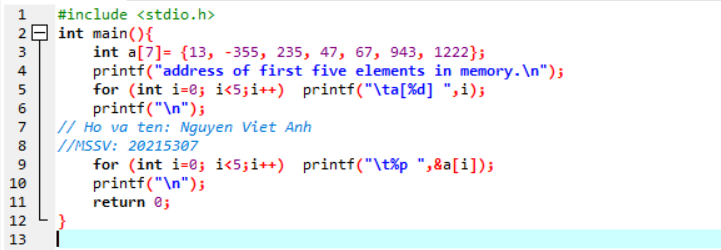
printf("z = %d", \*ptr);

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 Bài 1.2 In ra địa chỉ của phần tử trong mảng

#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");

// Ho va ten: Nguyen Viet Anh

//MSSV: 20215307

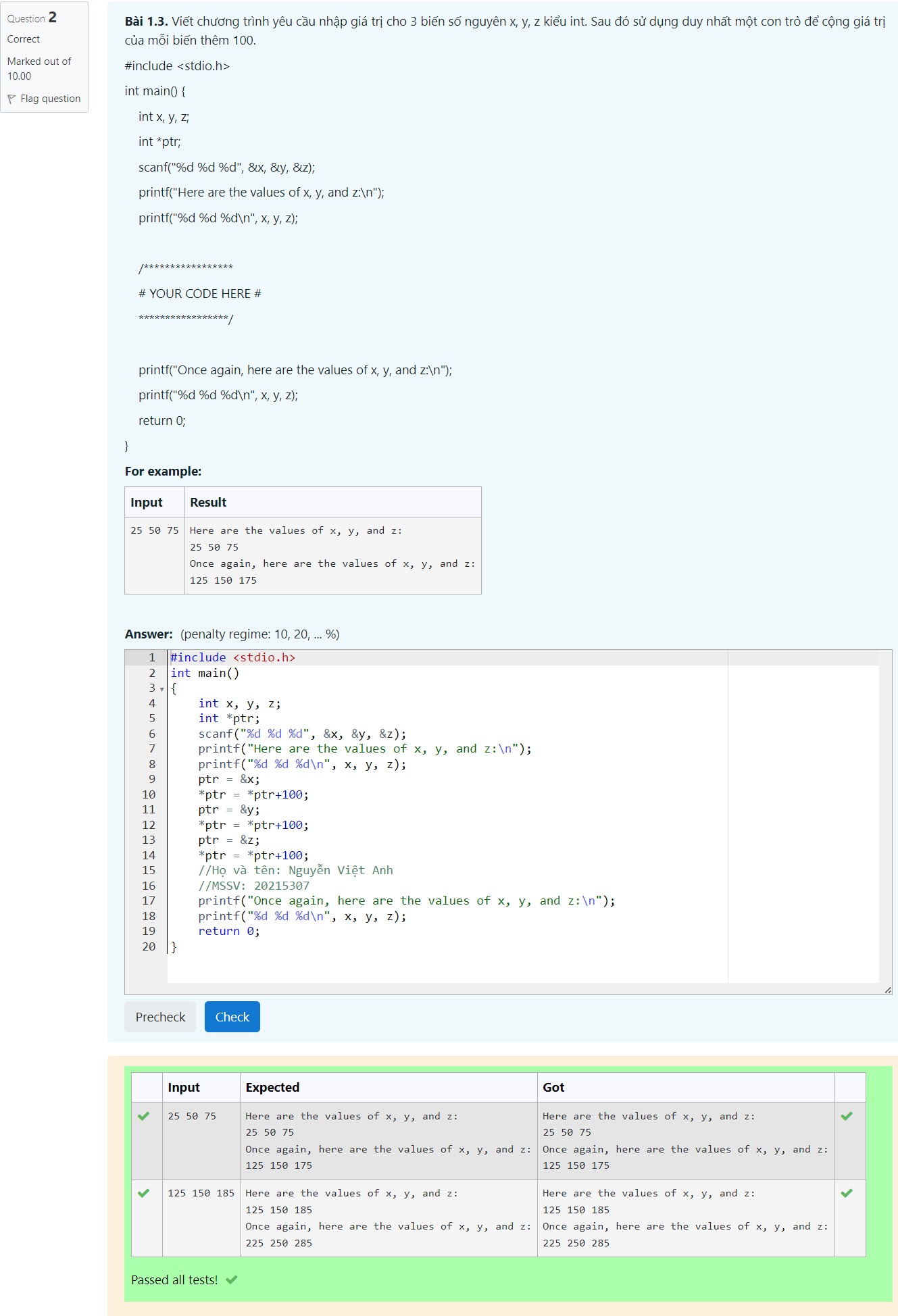
for (int i=0; i<5;i++) printf("\t%p ",&a[i]);

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.



#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);

ptr = &x;

\*ptr = \*ptr+100;

ptr = &y;

\*ptr = \*ptr+100;

ptr = &z;

\*ptr = \*ptr+100;

//Họ và tên: Nguyễn Việt Anh

//MSSV: 20215307

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

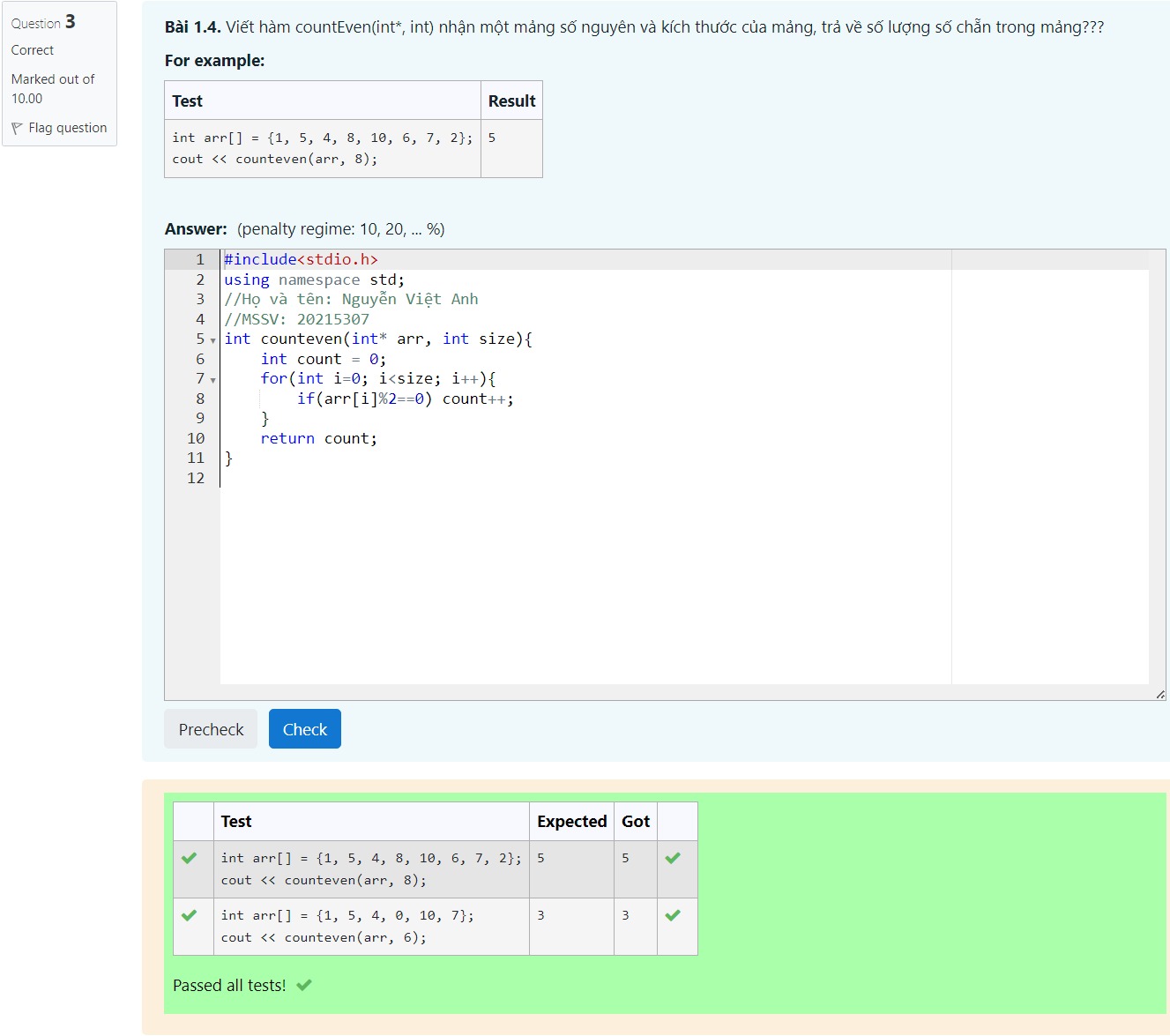
printf("%d %d %d\n", x, y, z);

return 0;

}

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

## **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.



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

#include<stdio.h>

using namespace std;

//Họ và tên: Nguyễn Việt Anh

//MSSV: 20215307

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

int count = 0;

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

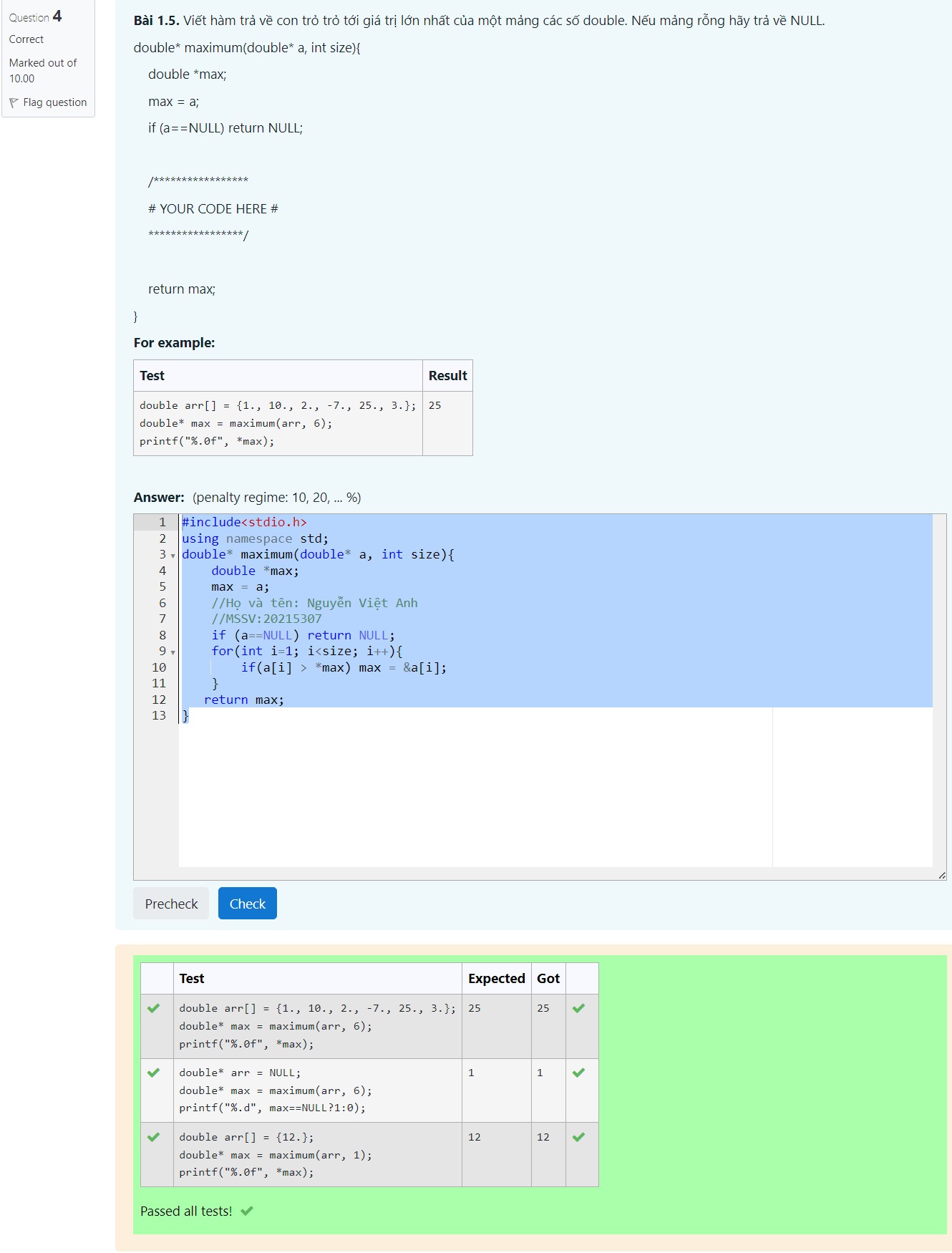
if(arr[i]%2==0) count++;

}

return count;

}

## **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.



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

#include<stdio.h>

using namespace std;

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

double \*max;

max = a;

//Họ và tên: Nguyễn Việt Anh

//MSSV:20215307

if (a==NULL) return NULL;

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

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

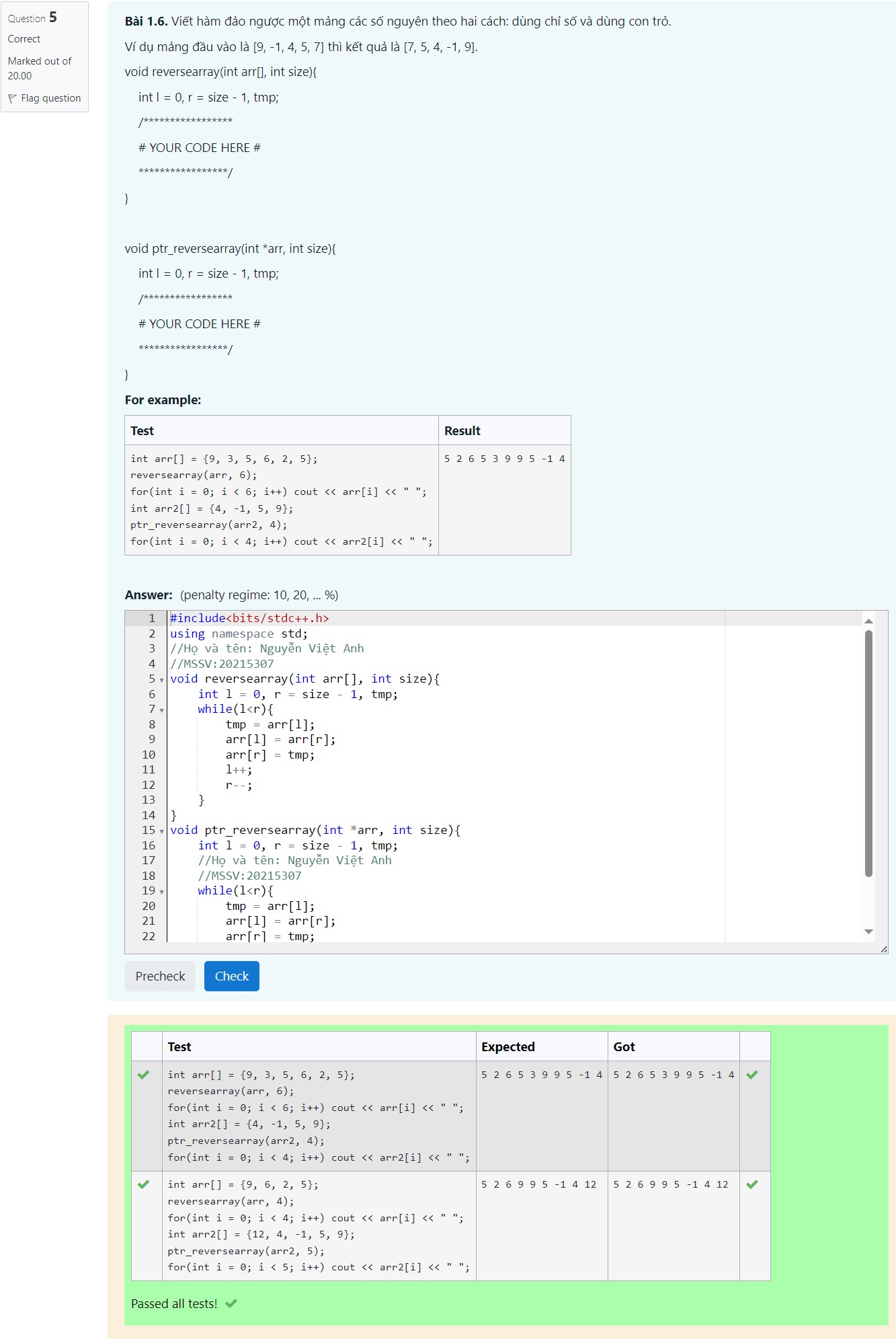
}

return max;

}

## **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 Hàm đảo ngược các phần tử trong mảng dùng chỉ số và con trỏ

#include<stdio.h>

using namespace std;

//Họ và tên: Nguyễn Việt Anh

//MSSV:20215307

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;

//Họ và tên: Nguyễn Việt Anh

//MSSV:20215307

while(l<r){

tmp = arr[l];

arr[l] = arr[r];

arr[r] = tmp;

l++;

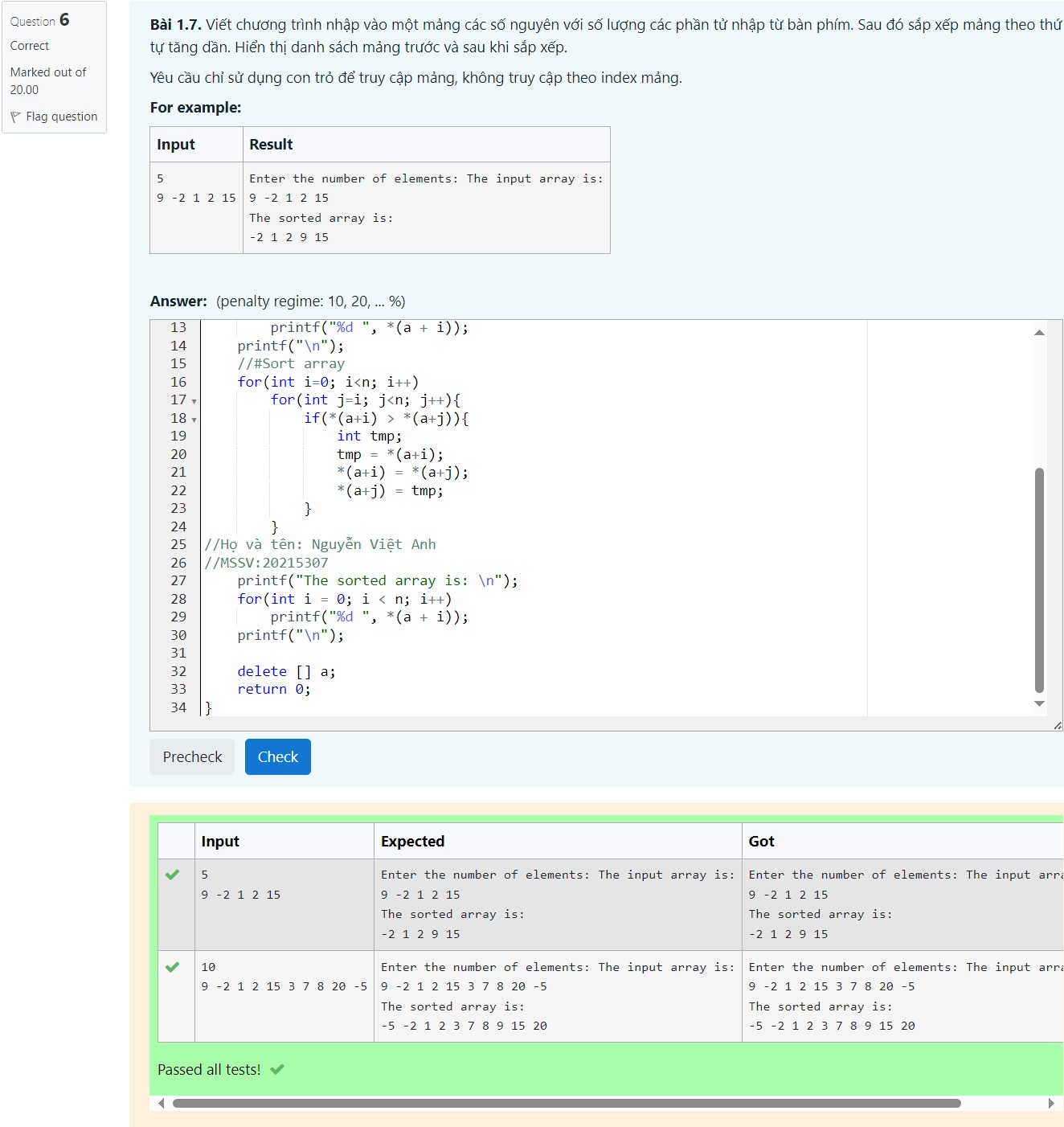
r--;

}

}

## **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.



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

#include <stdio.h>

int \*a;

int n, tmp;

int main(){

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

scanf("%d", &n);

//#Allocate memory

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;

}

}

//Họ và tên: Nguyễn Việt Anh

//MSSV:20215307

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

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

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

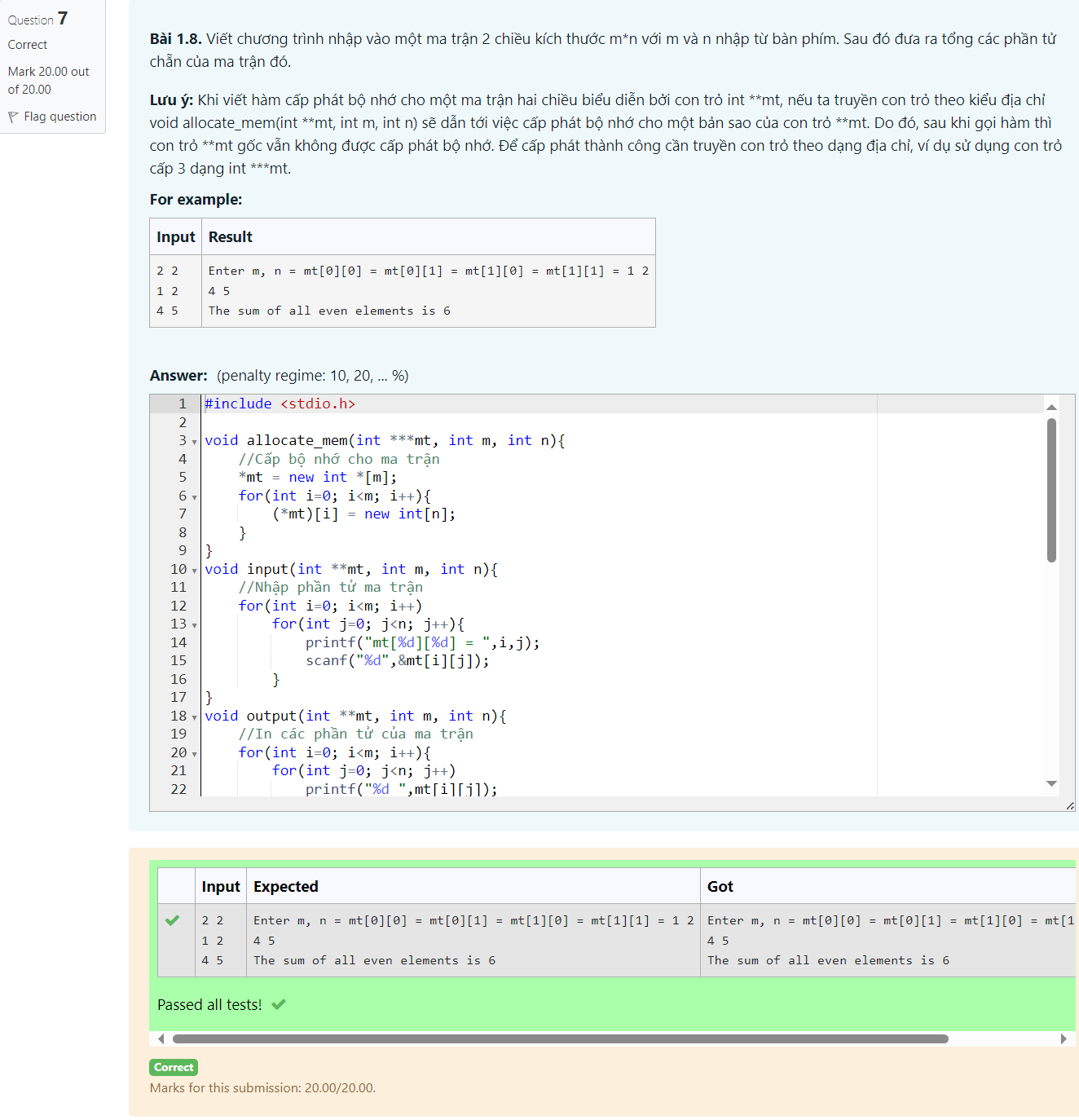
printf("\n");

delete [] a;

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.



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

#include <stdio.h>

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

//Cấp bộ nhớ cho ma trận

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

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

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

}

}

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

//Nhập phần tử ma trận

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){

//In các phần tử của ma trận

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;

//Tính tổng các phần tử ma trận

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;

}

//Họ và tên: Nguyễn Việt Anh

//MSSV:20215307

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

//Giải phóng bộ nhớ

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

delete []mt[i];

delete []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;

}

# **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.

#include<iostream>

using namespace std;

int main(){

int m;

cin >> m;

int d[m];

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

cin >> d[a];

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

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

for (int c = a; c<=b; c++)

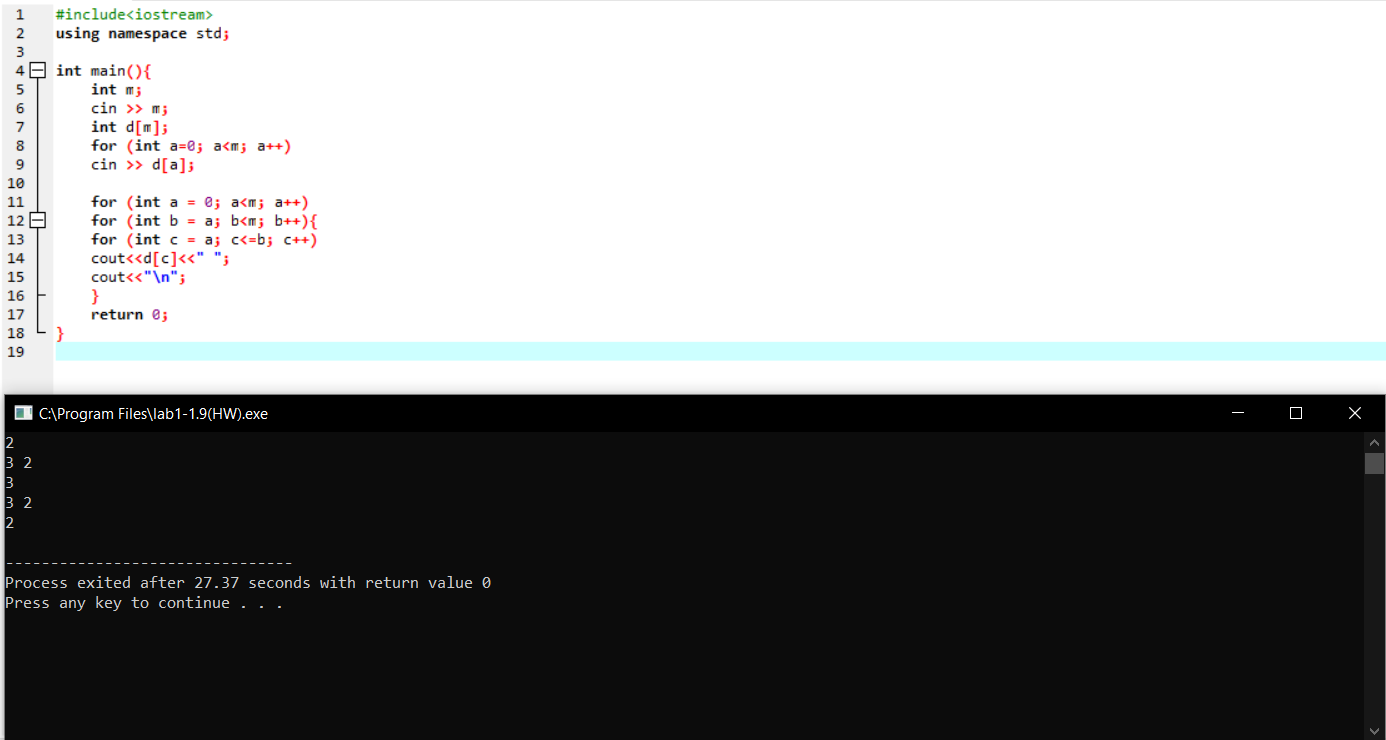
cout<<d[c]<<" ";

cout<<"\n";

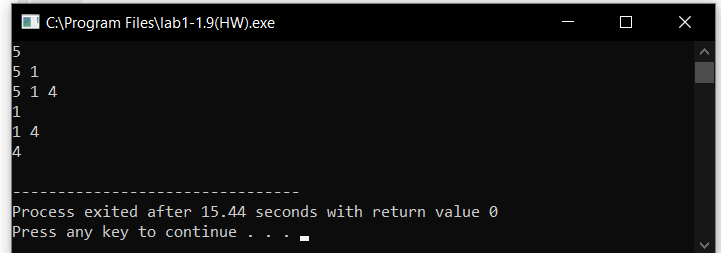
}

return 0;

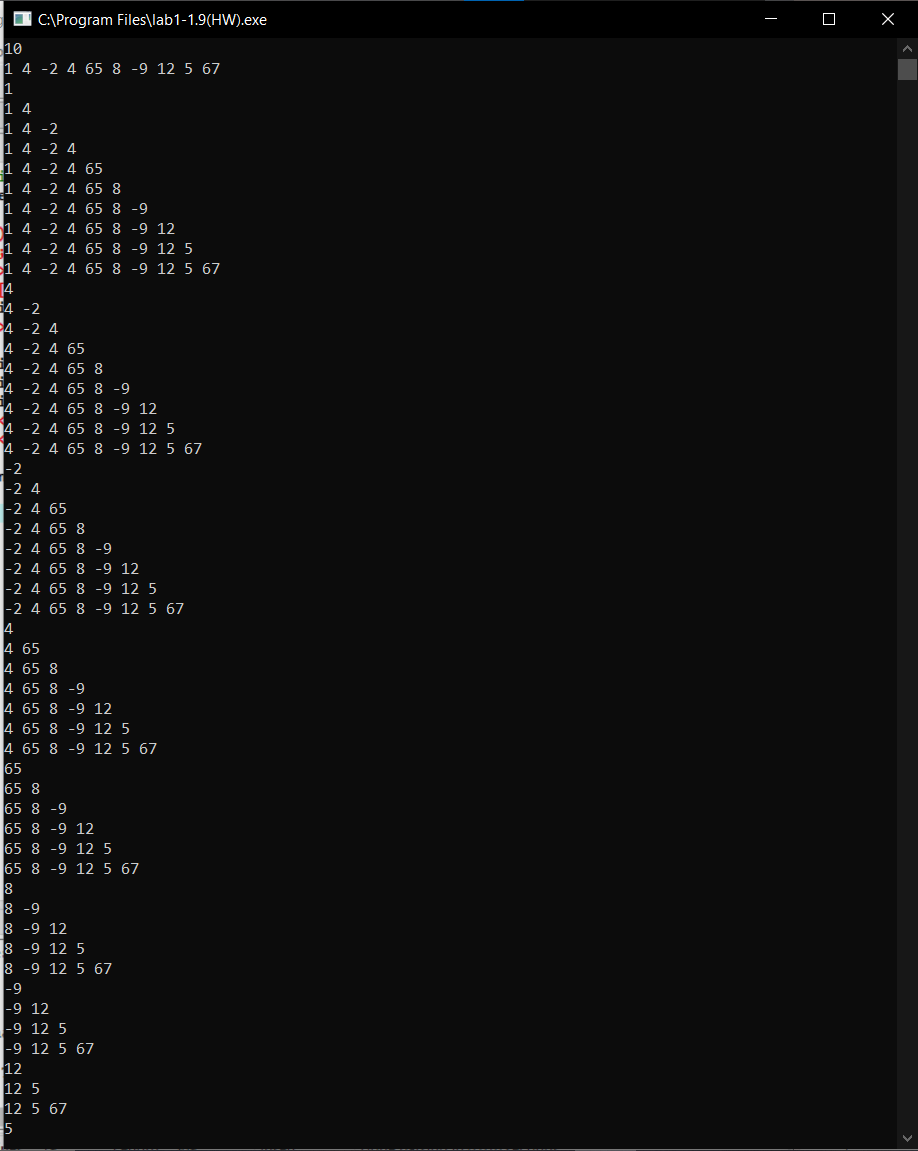
}



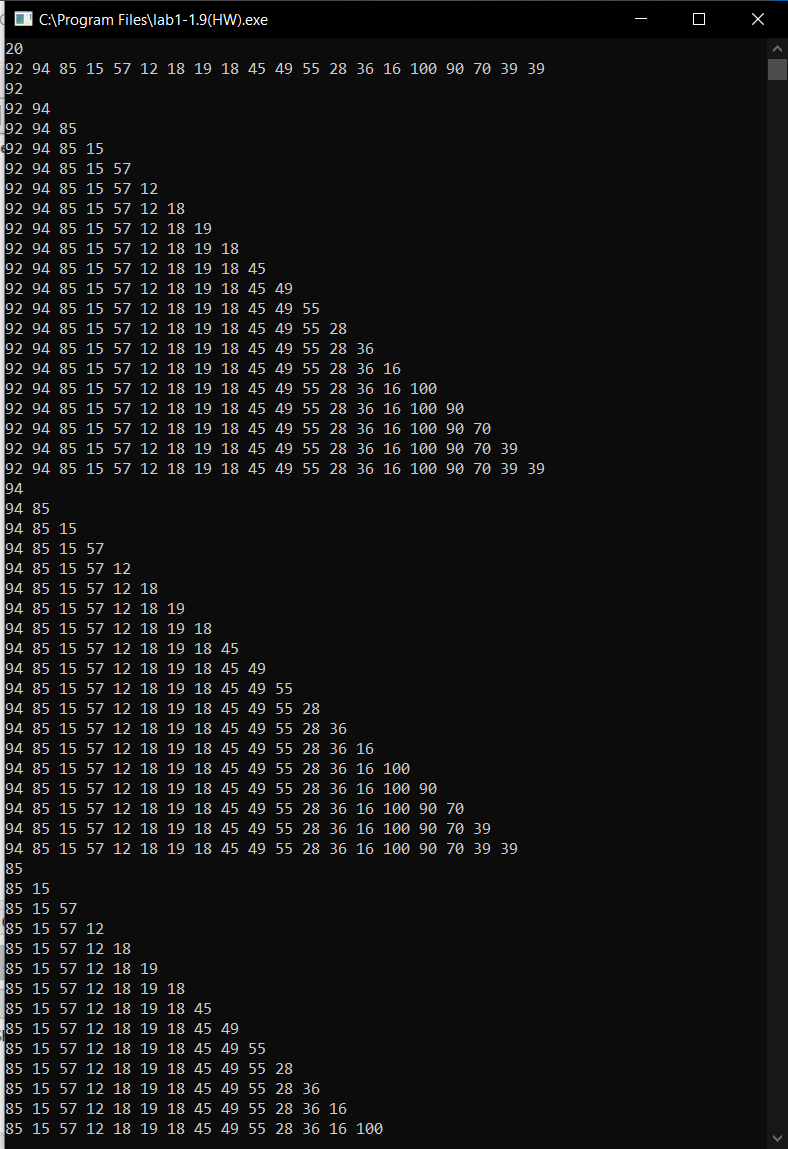
Case 1

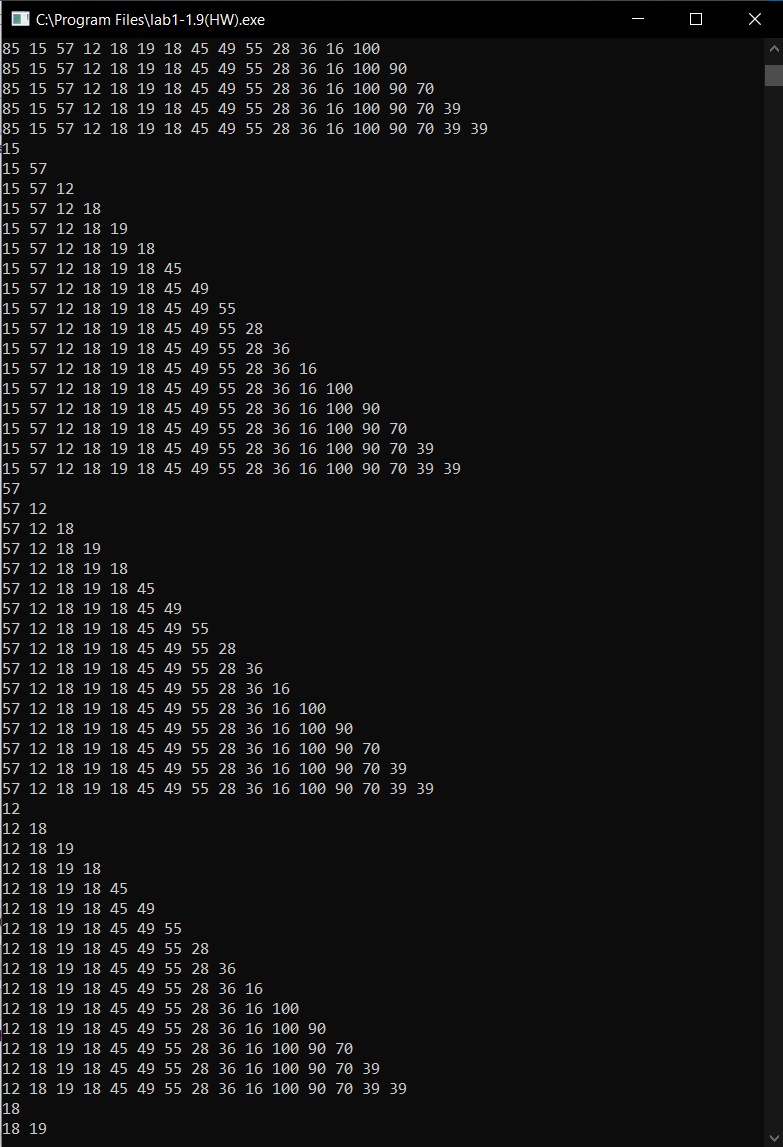


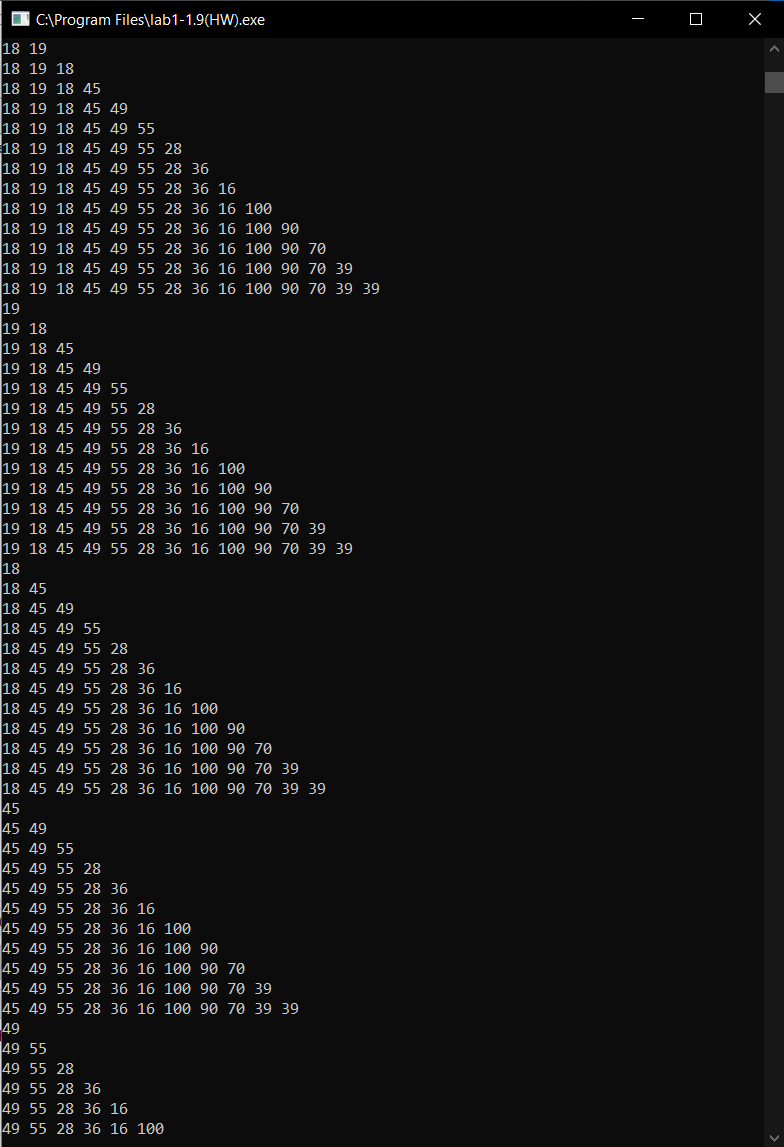
Case 2

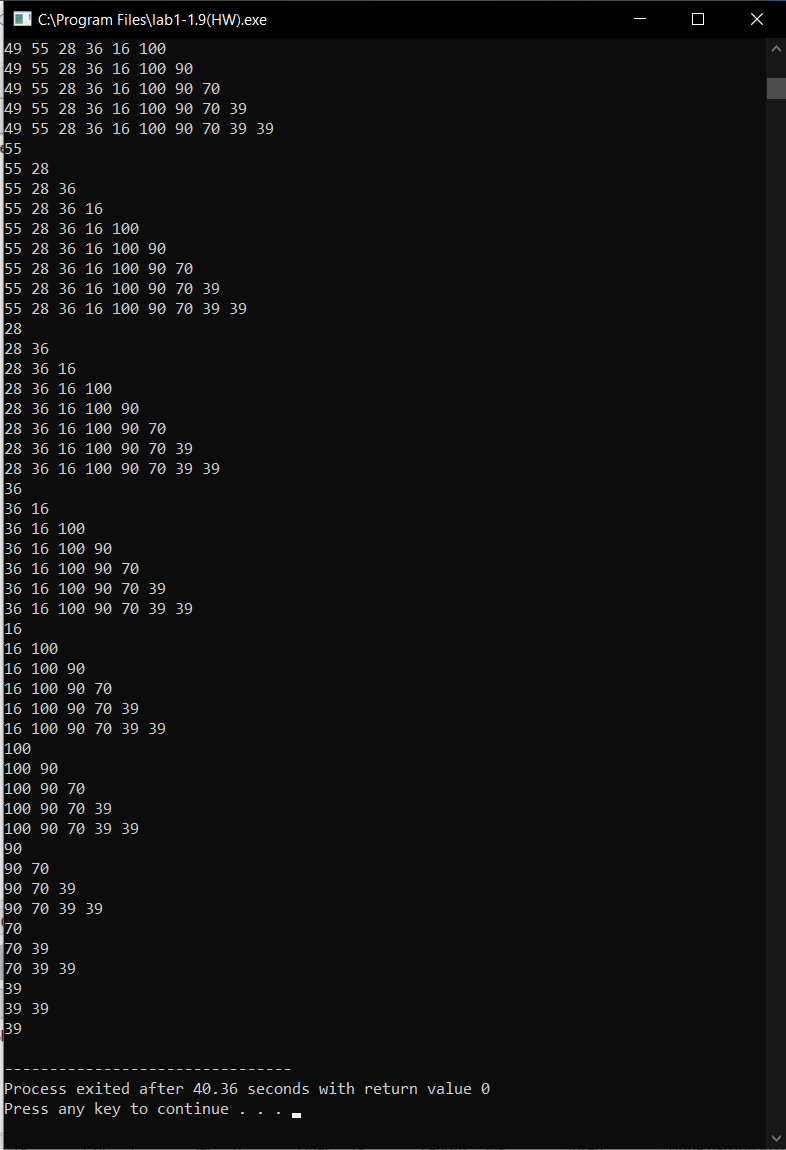


Case 3









Case 4

## **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.

#include<bits/stdc++.h>

using namespace std;

int main(){

int n;

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

cin >> n;

// Cap phat bo nho mtr1, mtr2

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 vao mtr1, mtr2

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 ra 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];

}

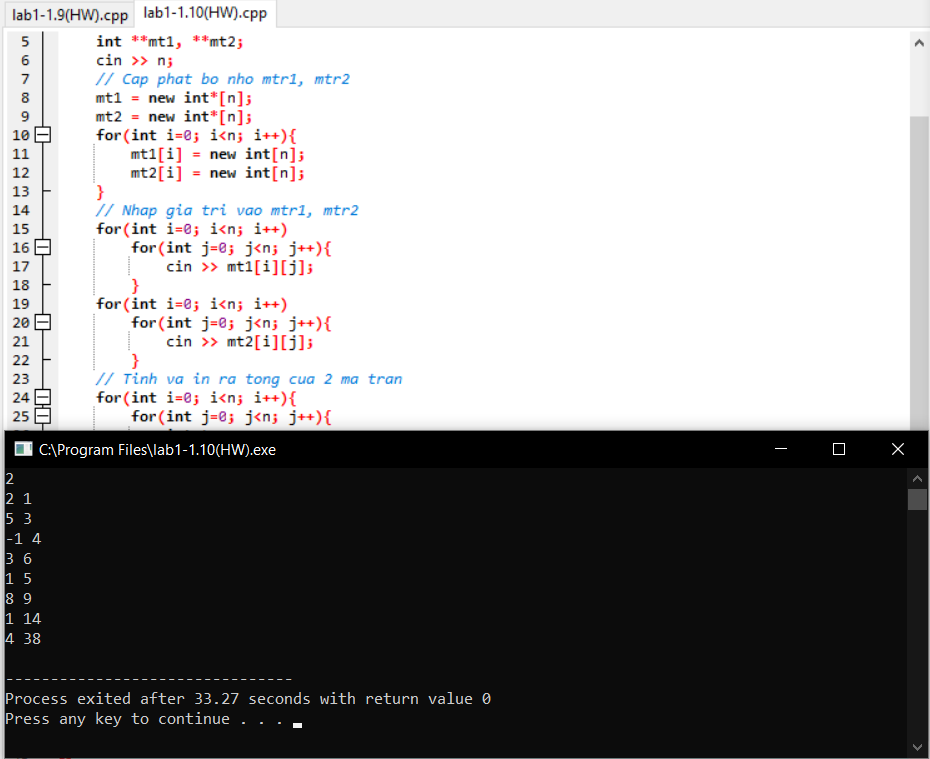
cout << temp << " ";

}

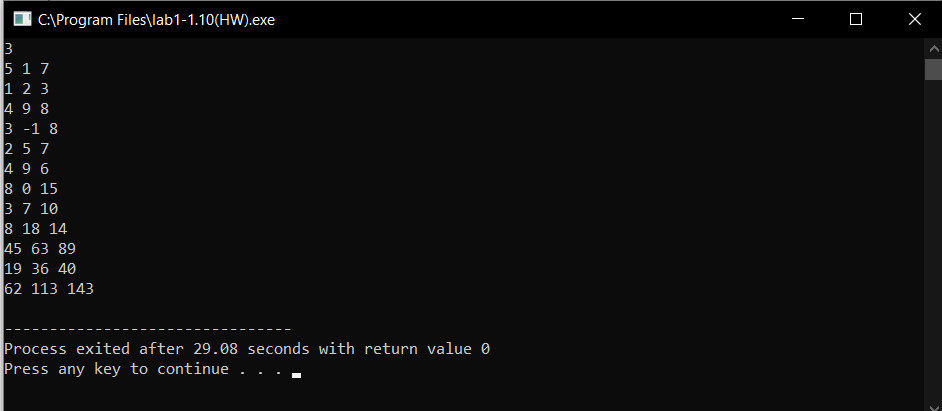
cout << endl;

}

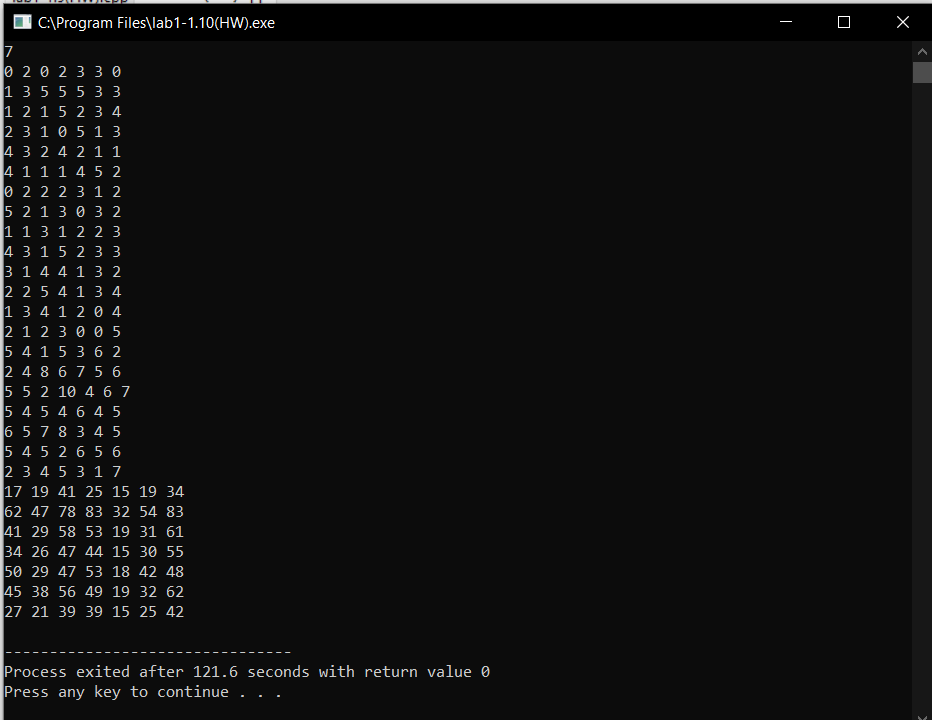
}



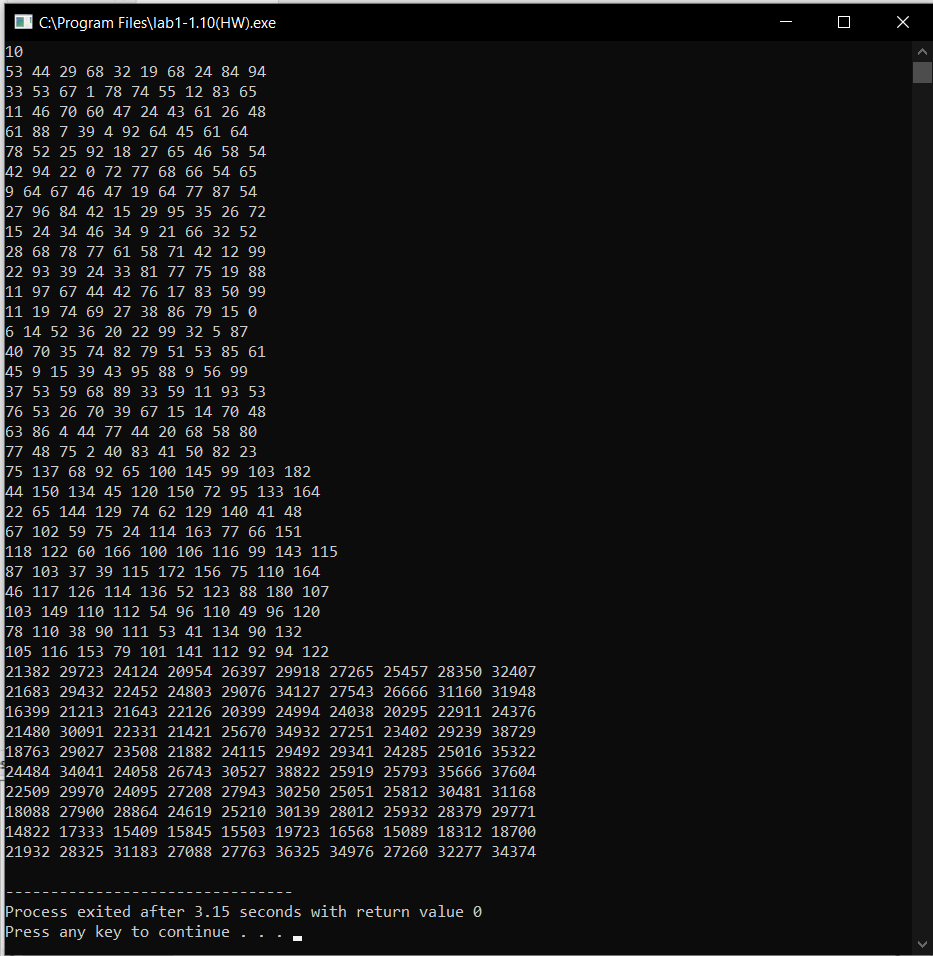
Case 1



Case 2



Case 3



Case 4