Contents

[Bài thực hành số 1 – Tuần 29 2](#_Toc68088527)

[Phần 1. Thực hành về con trỏ 2](#_Toc68088528)

[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. 2](#_Toc68088529)

[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 & 3](#_Toc68088530)

[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. 4](#_Toc68088531)

[Phần 2. Con trỏ và mảng 6](#_Toc68088532)

[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. 6](#_Toc68088533)

[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. 7](#_Toc68088534)

[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] 8](#_Toc68088535)

[Phần 3. Cấp phát động 10](#_Toc68088536)

[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. 10](#_Toc68088537)

[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. 13](#_Toc68088538)

[**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. 13](#_Toc68088539)

[Bài tập 9: Viết chương trình in ra tất cả các dãy con của một dãy cho trước. 16](#_Toc68088540)

[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. 18](#_Toc68088541)

# Bài thực hành số 1 – Tuần 29

## 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: 704713\_NguyenTienDat\_20194242\_Baitap1\_Buoi1.cpp

# include <stdio.h>

int main(){

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

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);

ptr = &y;

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

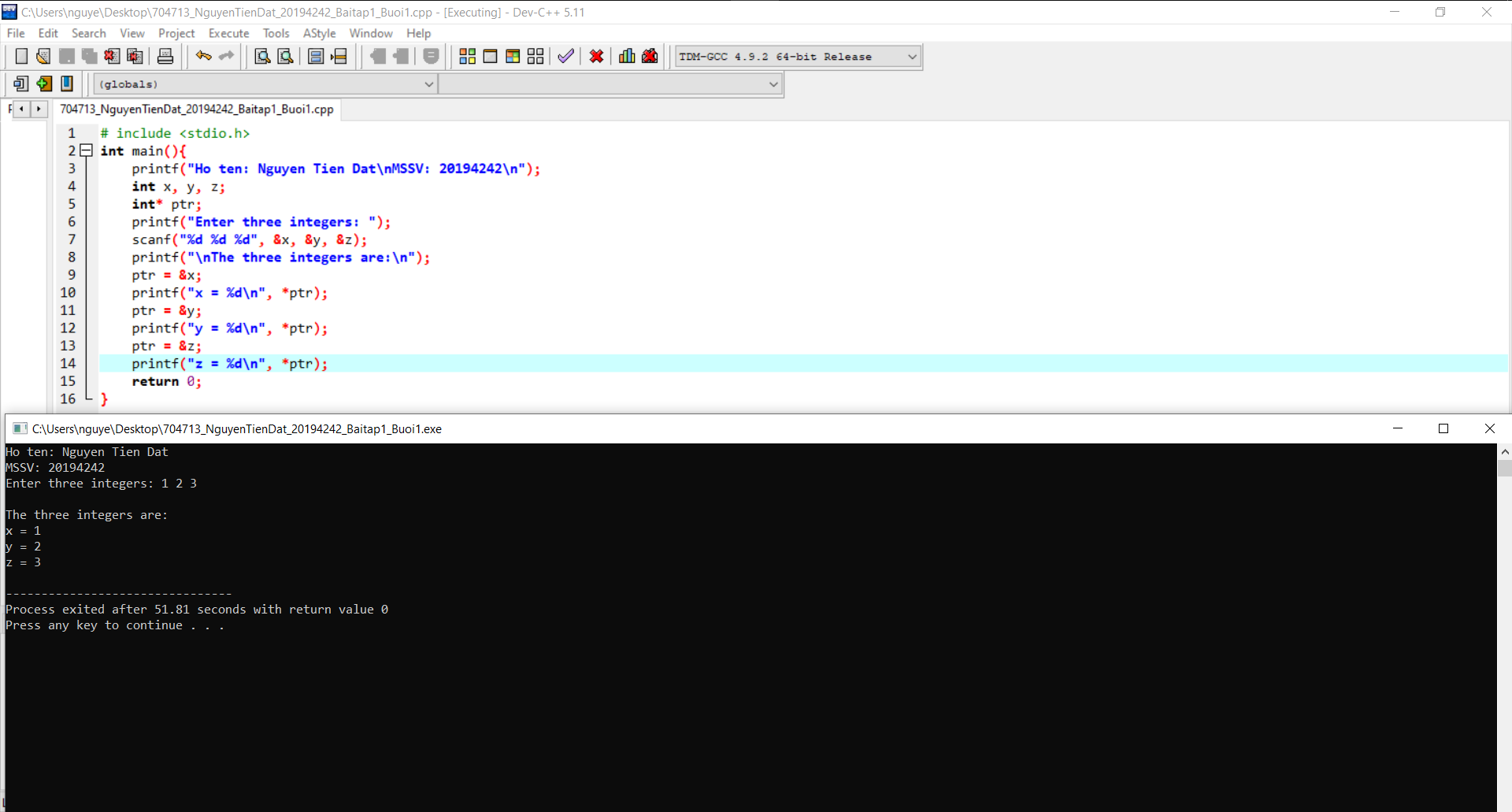
ptr = &z;

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

return 0;

}

Kết quả:



### 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: 704713\_NguyenTienDat\_20194242\_Baitap2\_Buoi1.cpp

#include <stdio.h>

int main(){

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\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("\t\ta[%d] ",i);

printf("\n");

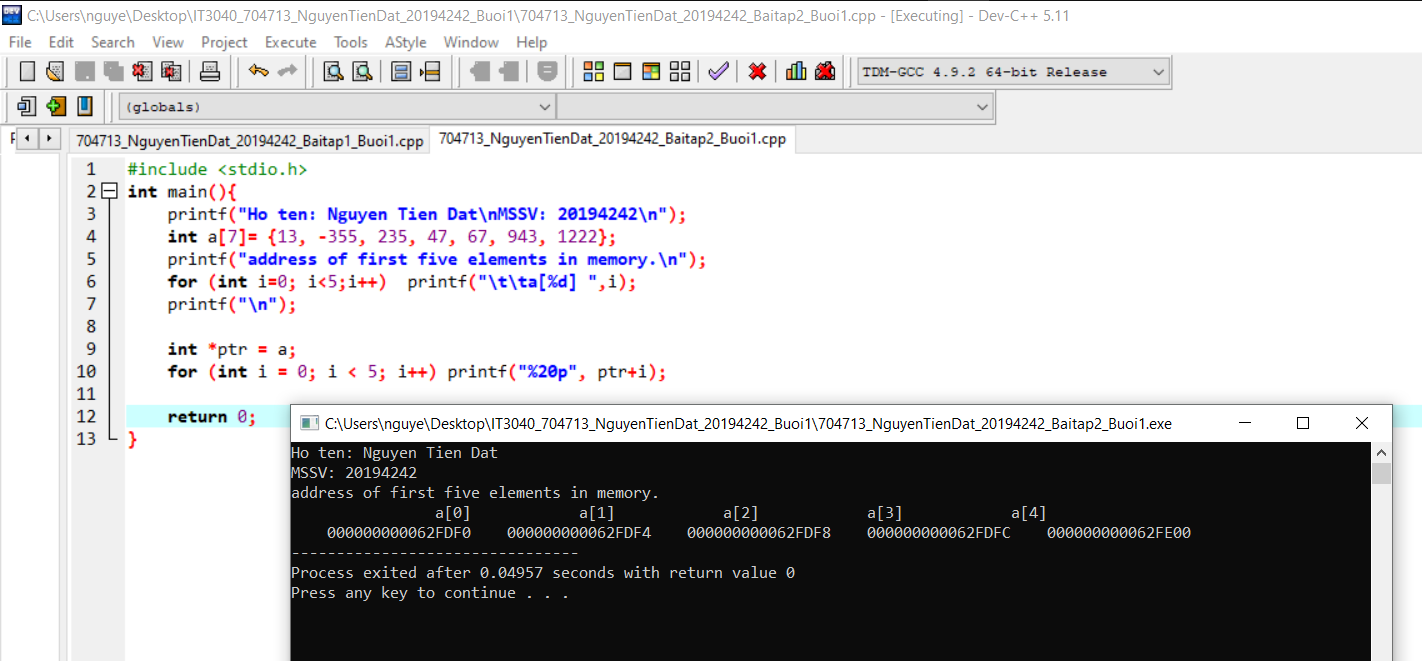
int \*ptr = a;

for (int i = 0; i < 5; i++) printf("%20p", ptr+i);

return 0;

}

Kết quả:



### 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: 704713\_NguyenTienDat\_20194242\_Baitap3\_Buoi1.cpp

#include <stdio.h>

int main()

{

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\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);

ptr = &x; \*ptr += 100;

ptr = &y; \*ptr += 100;

ptr = &z; \*ptr += 100;

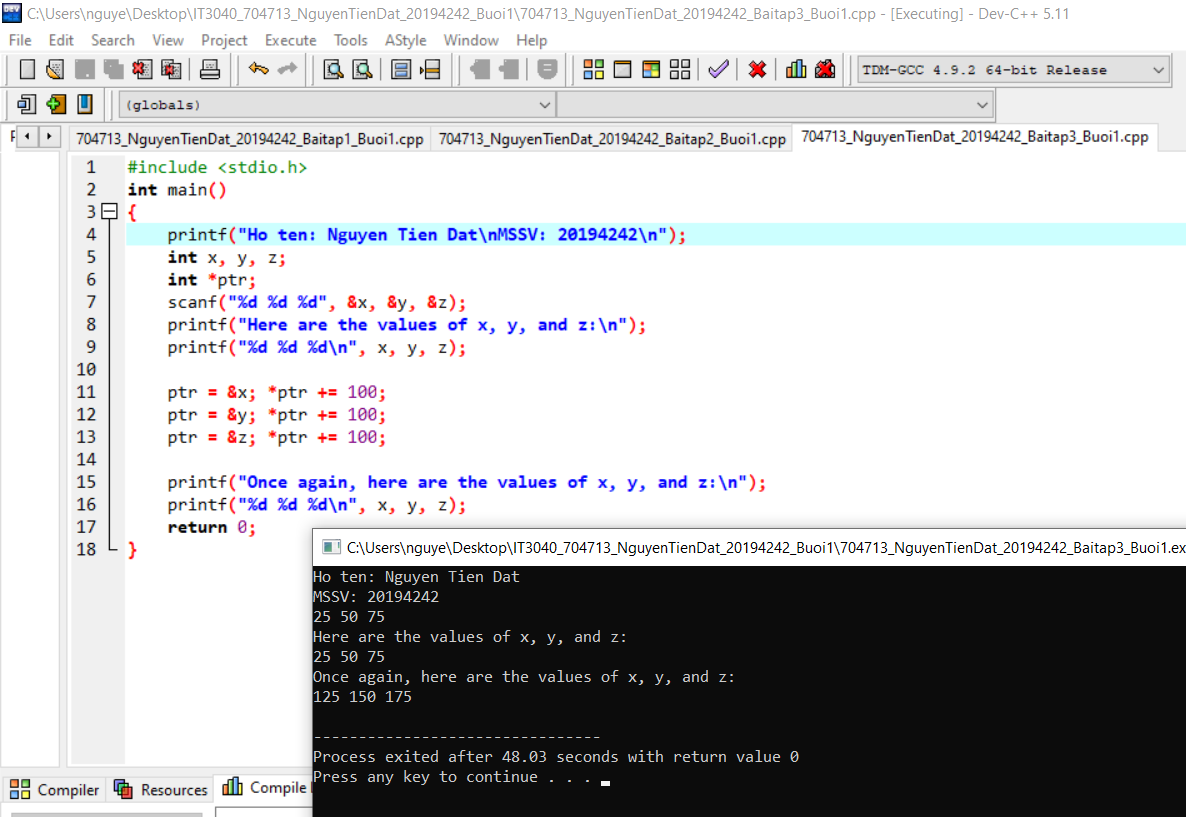
printf("Once again, here are the values of x, y, and z:\n");

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

return 0;

}

Kết quả:



## 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: 704713\_NguyenTienDat\_20194242\_Baitap4\_Buoi1.cpp

#include <stdio.h>

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

int count = 0;

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

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

}

return count;

}

int main() {

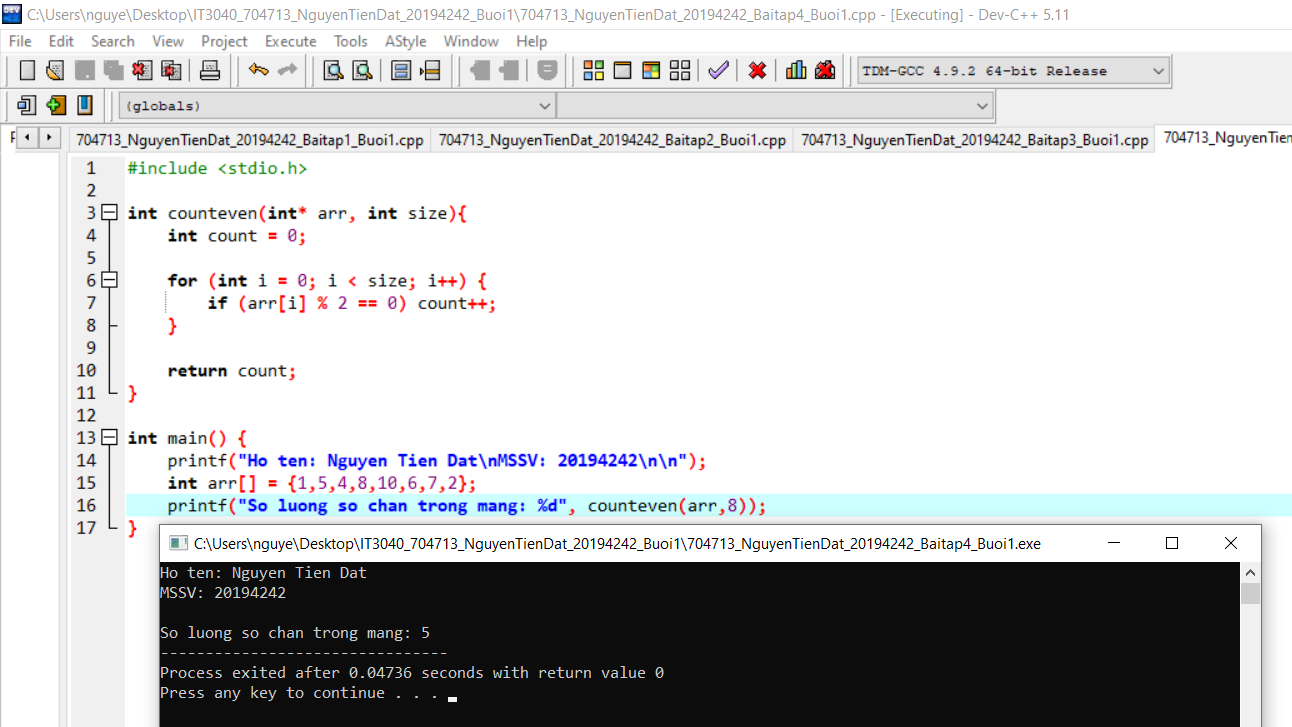
printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n\n");

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

printf("So luong so chan trong mang: %d", counteven(arr,8));

}

Kết quả:



## 

### 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: 704713\_NguyenTienDat\_20194242\_Baitap5\_Buoi1.cpp

#include <stdio.h>

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

double \*max;

max = a;

if (a==NULL) return NULL;

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

if (\*max < a[i]) max = a+i;

}

return max;

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n\n");

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

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

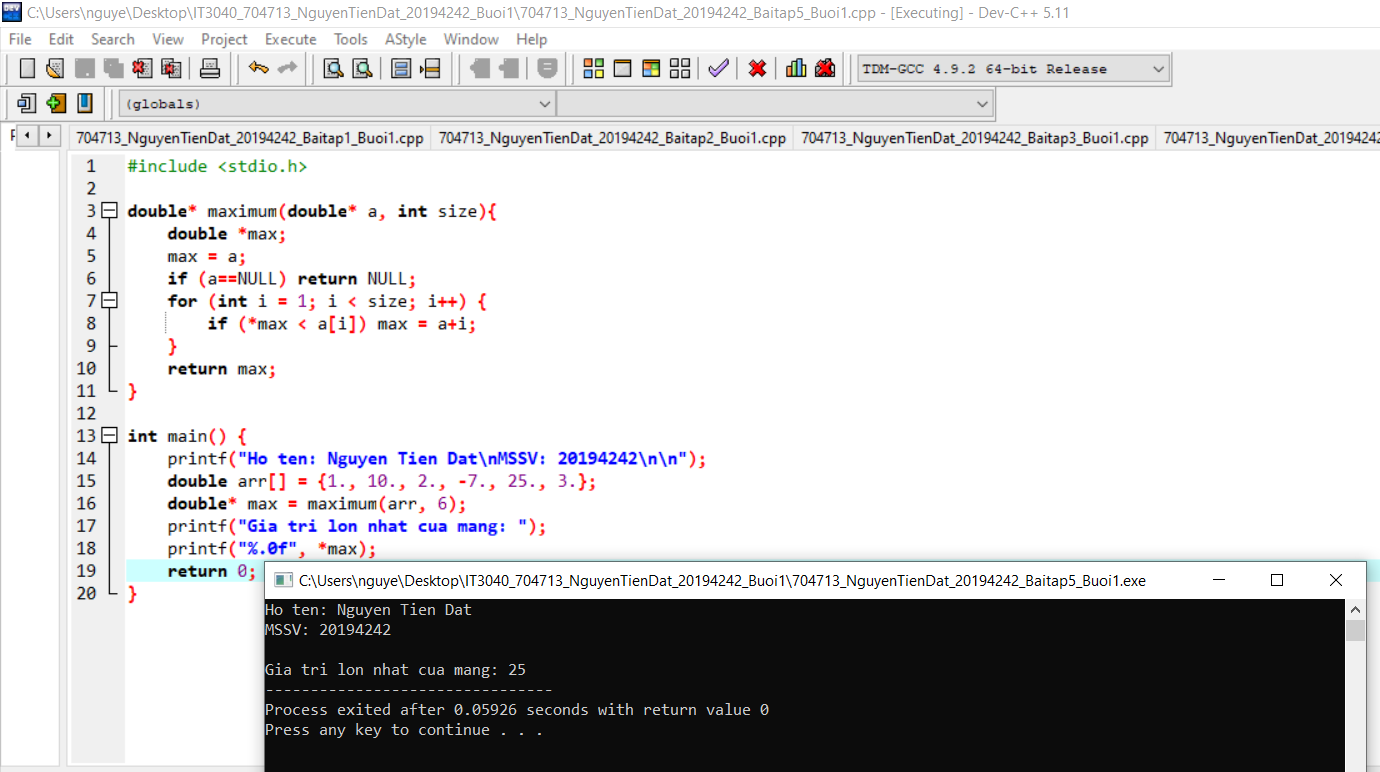
printf("Gia tri lon nhat cua mang: ");

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

return 0;

}

Kết quả:



### 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: 704713\_NguyenTienDat\_20194242\_Baitap6\_Buoi1.cpp

#include <stdio.h>

#include <iostream>

using namespace std;

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--;

}

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n\n");

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

cout << "Before reverse: " << endl;

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

reversearray(arr, 6);

cout << endl << "After reverse:" << endl;

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

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

cout << endl << "Before ptr\_reverse: " << endl;

for(int i = 0; i < 4; i++) cout << arr2[i] << " ";

ptr\_reversearray(arr2, 4);

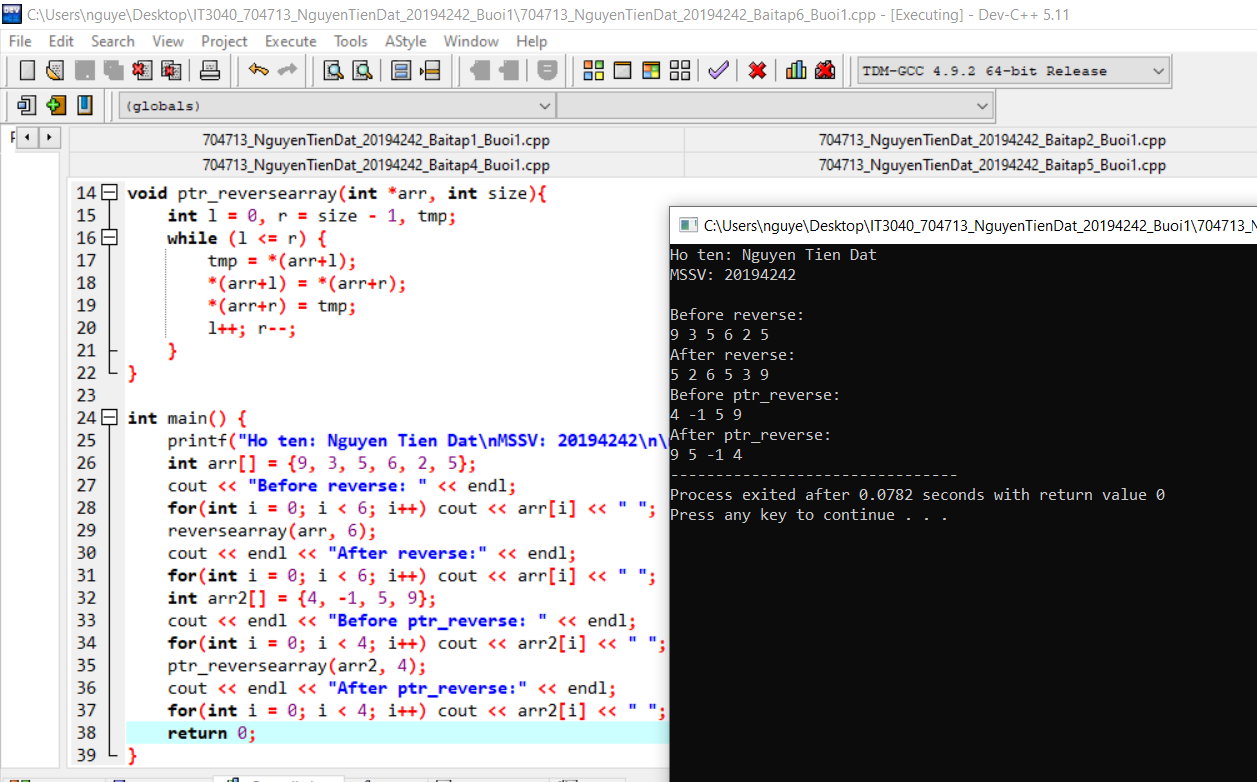
cout << endl << "After ptr\_reverse:" << endl;

for(int i = 0; i < 4; i++) cout << arr2[i] << " ";

return 0;

}

Kết quả:



## 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: 704713\_NguyenTienDat\_20194242\_Baitap7\_Buoi1.cpp

#include <stdio.h>

int \*a;

int n, tmp;

int main(){

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n\n");

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+1; j < n; j++) {

if (a[i] > a[j]) {

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));

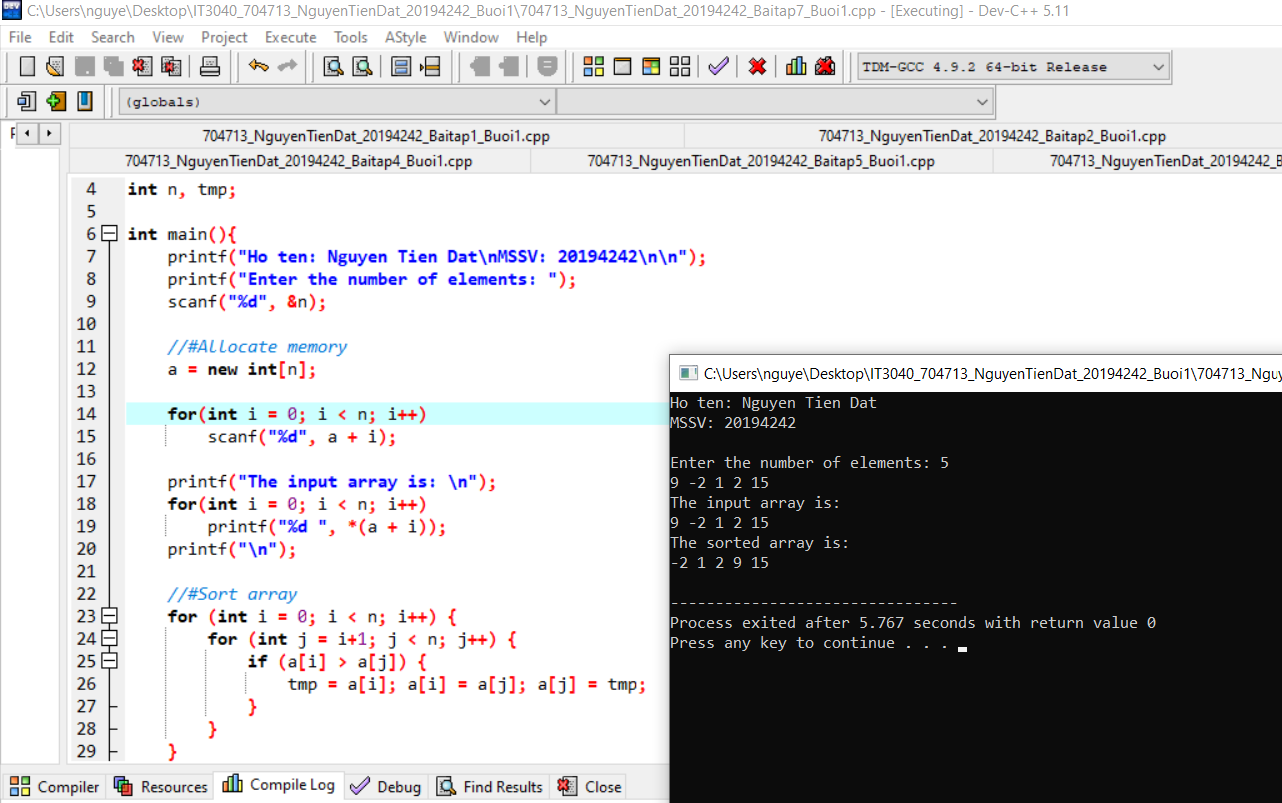
printf("\n");

delete [] a;

return 0;

}

Kết quả:



### 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: 704713\_NguyenTienDat\_20194242\_Baitap8\_Buoi1.cpp

#include <stdio.h>

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

//#Allocate memory for the matrix

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

for (int i = 0; i < n; i++) (\*mt)[i] = new int[n];

}

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

//#Input elements of the matrix

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

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

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

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

}

}

}

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

int tong = 0;

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

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

for (int i = 0; i < m; i++) delete[] mt[i];

delete[] mt;

}

int main(){

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n\n");

int m, n, \*\*mt;

printf("Enter m, n = ");

scanf("%d%d", &m, &n);

allocate\_mem(&mt, m, n);

input(mt, m, n);

printf("The elements of matrix: \n");

output(mt, m, n);

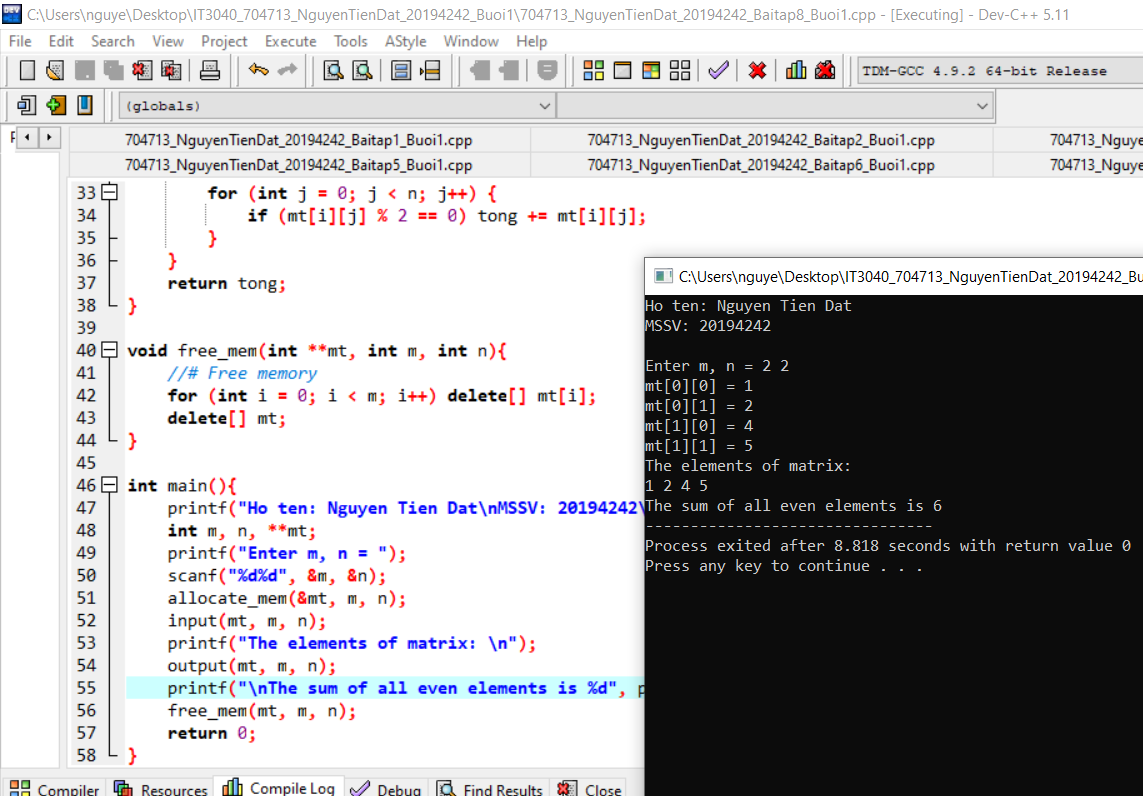
printf("\nThe sum of all even elements is %d", process(mt, m, n));

free\_mem(mt, m, n);

return 0;

}

Kết quả:



### Bài tập 9: Viết chương trình in ra tất cả các dãy con của một dãy cho trước.

Tên file: 704713\_NguyenTienDat\_20194242\_Baitap9\_Buoi1.cpp

#include <stdio.h>

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n\n");

// day cho truoc:

int arr[] = {1,3,4,2};

int n = 4;

printf("Day cho truoc: \n");

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

printf("\nCac day con: \n");

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

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

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

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

}

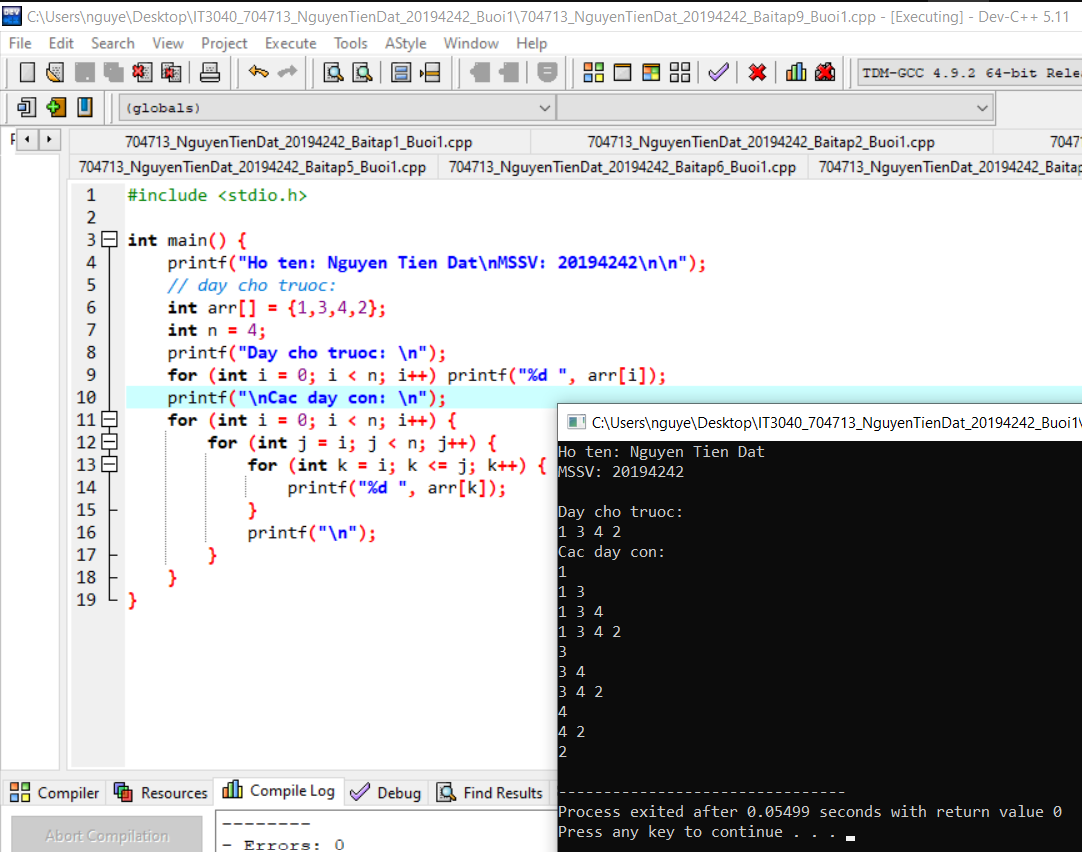
printf("\n");

}

}

}

Kết quả:



### 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: 704713\_NguyenTienDat\_20194242\_Baitap10\_Buoi1.cpp

#include <stdio.h>

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

//#Allocate memory for the matrix

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

for (int i = 0; i < n; i++) (\*mt)[i] = new int[n];

}

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

//#Input elements of the matrix

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

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

printf("%c[%d][%d] = ",name, i ,j);

scanf("%d", (\*(mt + i) + j));

}

}

}

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

//# Print all elements of the matrix

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

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

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

}

printf("\n");

}

}

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

//# Free memory

for (int i = 0; i < m; i++) delete[] mt[i];

delete[] mt;

}

int main() {

printf("Hoten: Nguyen Tien Dat\nMSSV: 20194242\n\n");

int n, \*\*A, \*\*B;

printf("Enter the size of matrix n = "); scanf("%d", &n);

allocate\_mem(&A,n,n);

allocate\_mem(&B,n,n);

printf("Input data to matrix:\n");

input(A,n,n,'A');

input(B,n,n,'B');

printf("Matrix A:\n"); output(A,n,n);

printf("Matrix B:\n"); output(B,n,n);

// process:

// Tong 2 matran: A + B

printf("A + B = \n");

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

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

printf("%d ", A[i][j] + B[i][j]);

}

printf("\n");

}

// Tich 2 matran:

printf("A \* B = \n");

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

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

int tmp = 0;

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

tmp += A[i][k] \* B[k][j];

}

printf("%d ", tmp);

}

printf("\n");

}

}

Kết quả:

