**Data Structure LAB TASK 1:**



**Submitted by:**

Name: Anum Batool

Roll No: Sp22-BCS-112

Section: B

**Submitted to:**

Mam Yasmeen Jana

**Activity 1:**

**Programs of Pointers:**

**Program # 1:**

#include <iostream>

using namespace std;

int main() {

int x = 10;

int\* ptr = &x; // Declare and initialize a pointer to an integer

std::cout << "Value of x: " << x << std::endl;

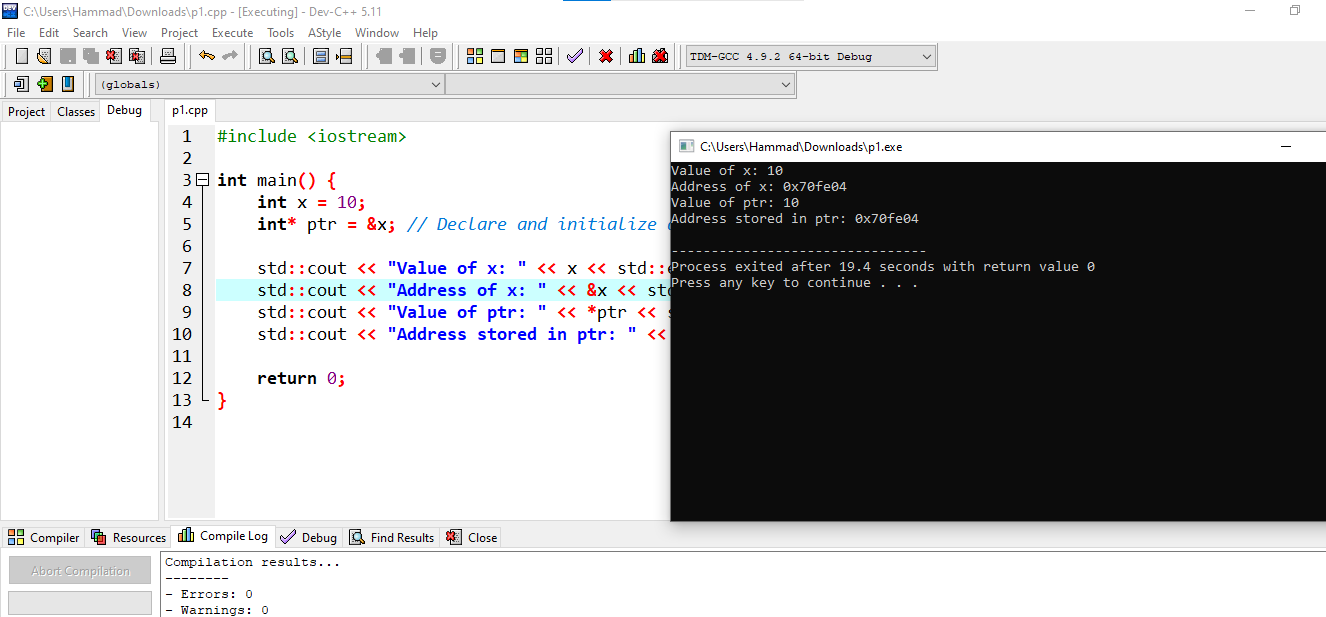
std::cout << "Address of x: " << &x << std::endl;

std::cout << "Value of ptr: " << \*ptr << std::endl; // Dereferencing the pointer

std::cout << "Address stored in ptr: " << ptr << std::endl;

return 0;

}



**Program # 2:**

#include <iostream>

Using namespace std;

void modifyValue(int\* ptr) {

(\*ptr)++; // Increment the value pointed to by ptr

}

int main() {

int x = 5;

int\* ptr = &x;

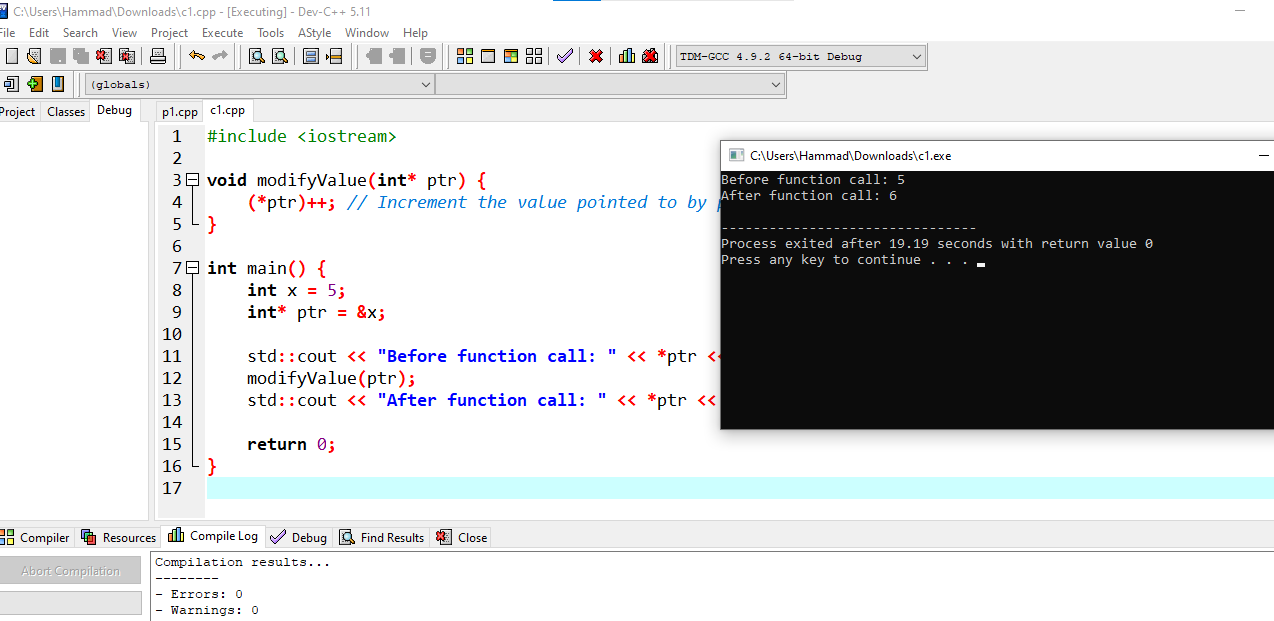
std::cout << "Before function call: " << \*ptr << std::endl;

modifyValue(ptr);

std::cout << "After function call: " << \*ptr << std::endl;

return 0;

}



**Program # 3:**

#include <iostream>

Using namespace std;

int main() {

int x = 10, y = 20, z = 30;

int\* arr[] = {&x, &y, &z}; // Array of pointers to integers

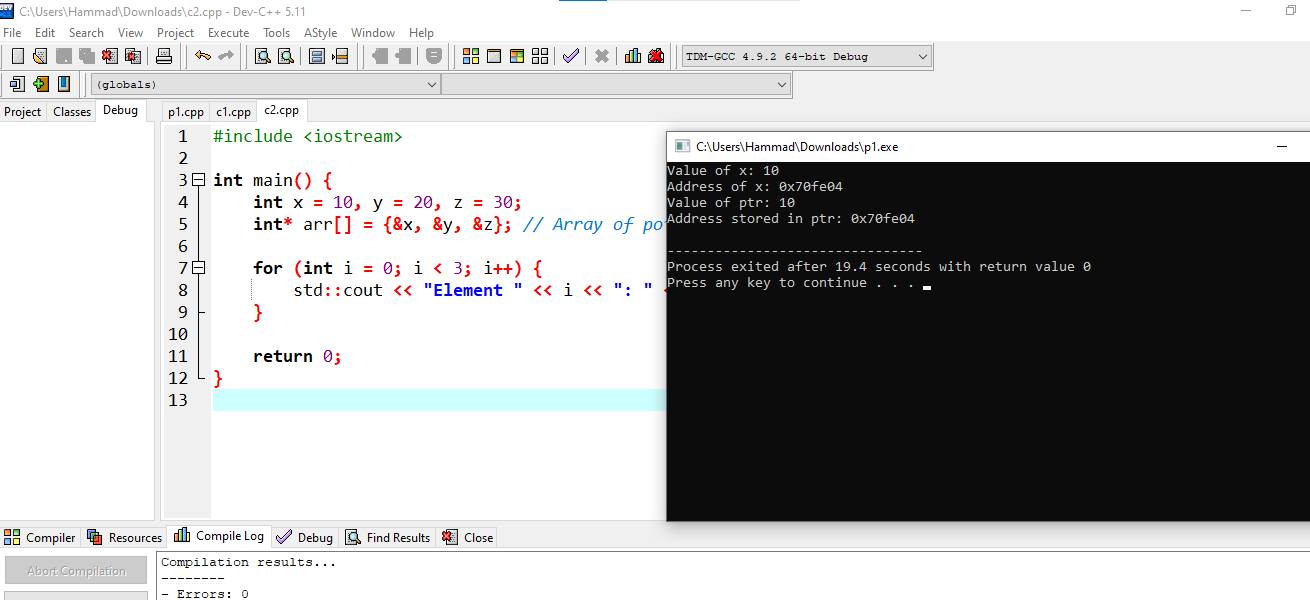
for (int i = 0; i < 3; i++) {

std::cout << "Element " << i << ": " << \*(arr[i]) << std::endl;

}

return 0;

}



**Program #4:**

#include <iostream>

Using namespace std;

struct Point {

int x;

int y;

};

int main() {

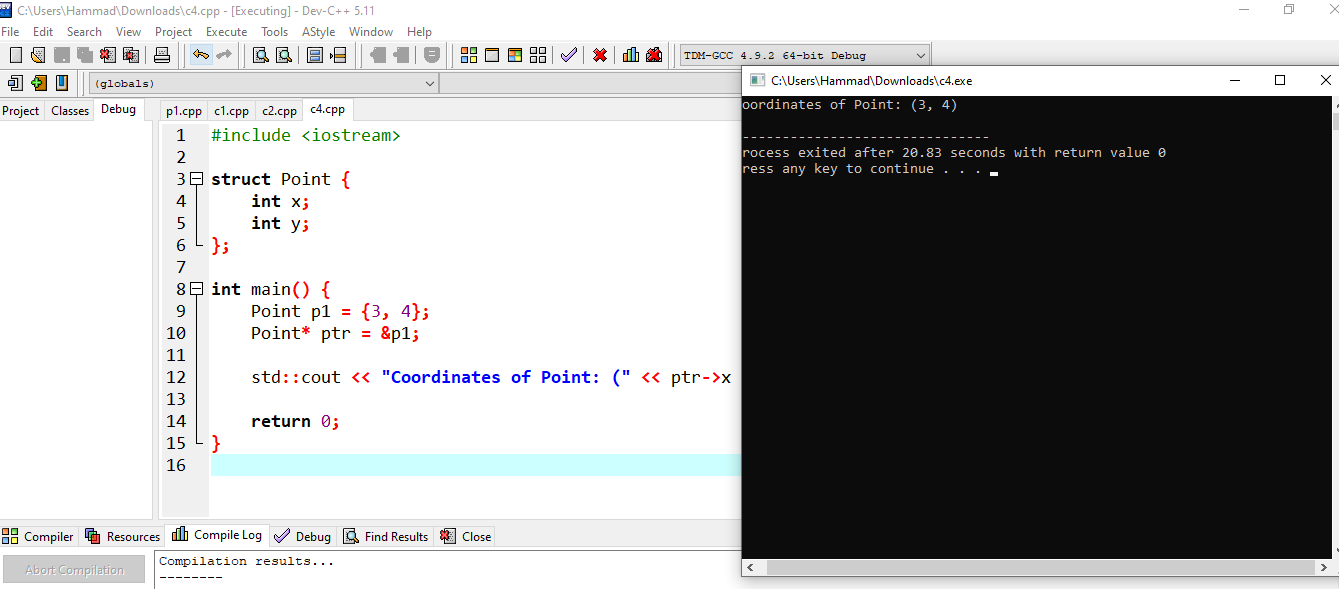
Point p1 = {3, 4};

Point\* ptr = &p1;

std::cout << "Coordinates of Point: (" << ptr->x << ", " << ptr->y << ")" << std::endl;

return 0;

}



**Program #5:**

#include <iostream>

Using namespace std;

int main() {

int x = 10;

int\* ptr1 = &x;

int\*\* ptr2 = &ptr1; // Pointer to a pointer

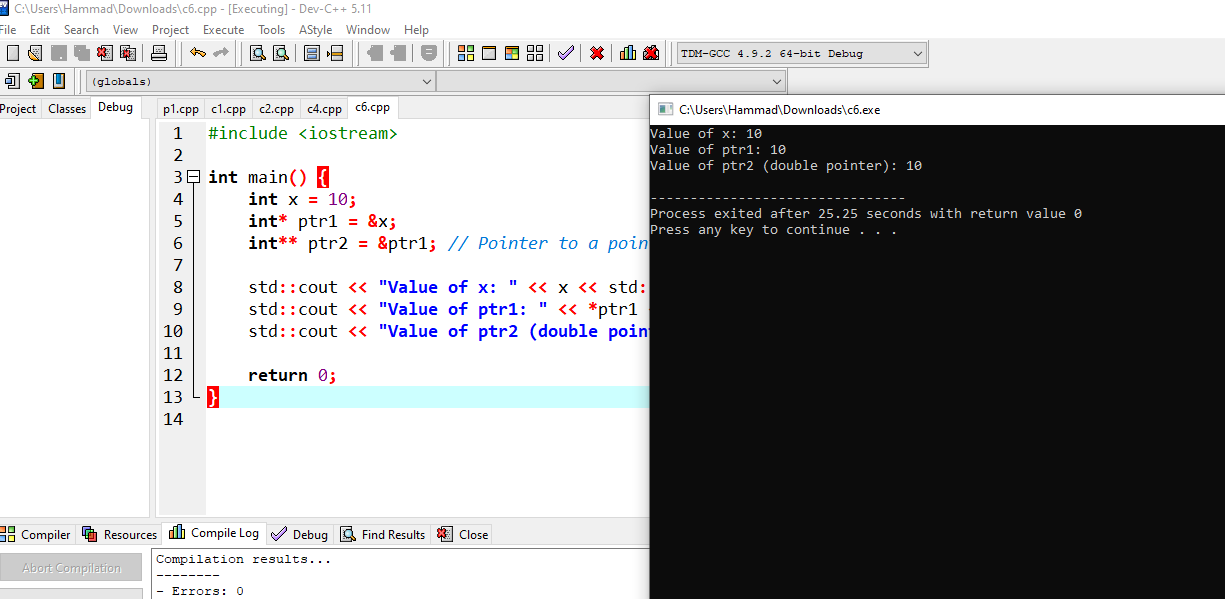
std::cout << "Value of x: " << x << std::endl;

std::cout << "Value of ptr1: " << \*ptr1 << std::endl;

std::cout << "Value of ptr2 (double pointer): " << \*\*ptr2 << std::endl;

return 0;

}



**Program #6:**

#include <iostream>

Using namespace std;

int main() {

int n;

std::cout << "Enter the number of elements: ";

std::cin >> n;

int\* arr = new int[n]; // Dynamically allocate an array of integers

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

arr[i] = i \* 2; // Initialize the elements

}

std::cout << "Array elements: ";

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

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

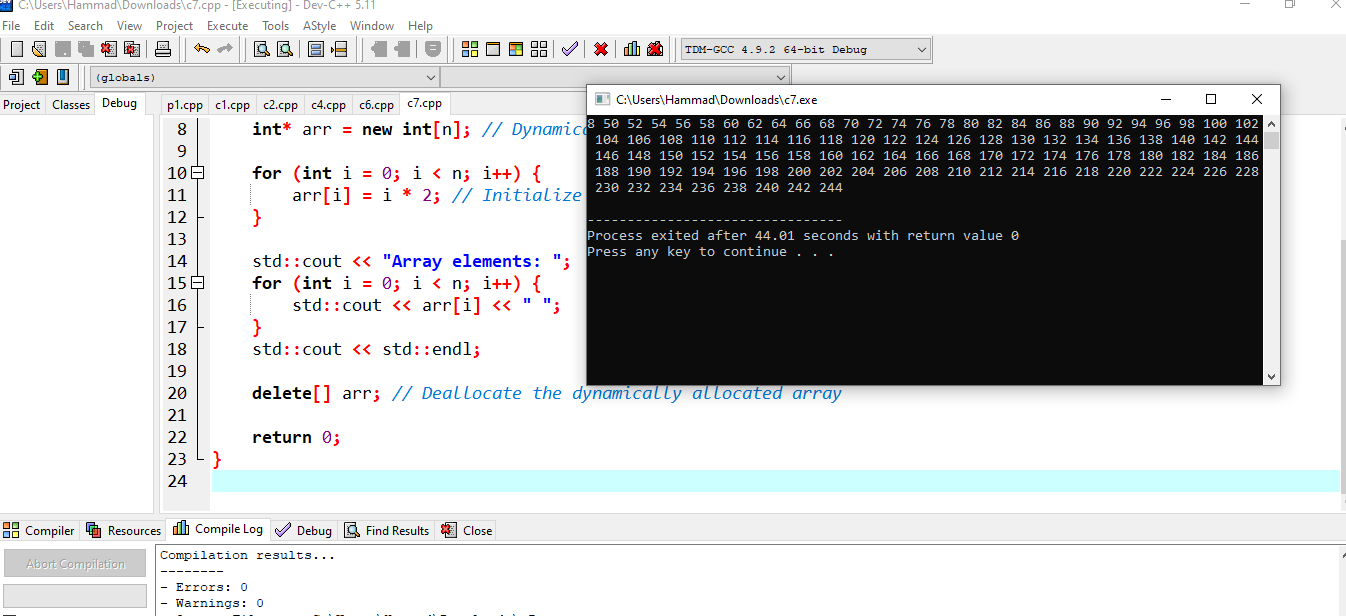
}

std::cout << std::endl;

delete[] arr; // Deallocate the dynamically allocated array

return 0;

}



**Program #7:**

#include <iostream>

Using namespace std;

int main() {

int x = 10;

const int y = 20;

int\* ptr1 = &x; // Pointer to a non-constant integer

const int\* ptr2 = &x; // Pointer to a constant integer

int\* const ptr3 = &x; // Constant pointer to a non-constant integer

const int\* const ptr4 = &y; // Constant pointer to a constant integer

std::cout << "Value pointed by ptr1: " << \*ptr1 << std::endl;

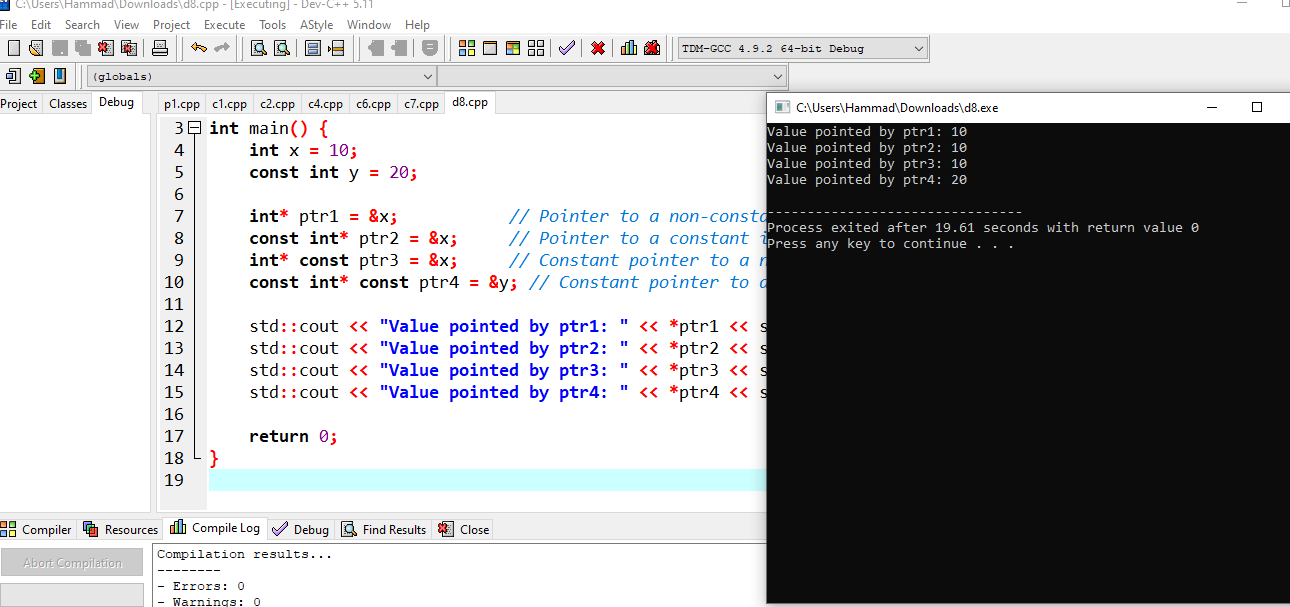
std::cout << "Value pointed by ptr2: " << \*ptr2 << std::endl;

std::cout << "Value pointed by ptr3: " << \*ptr3 << std::endl;

std::cout << "Value pointed by ptr4: " << \*ptr4 << std::endl;

return 0;

}



**Program#8:**

#include <iostream>

Using namespace std;

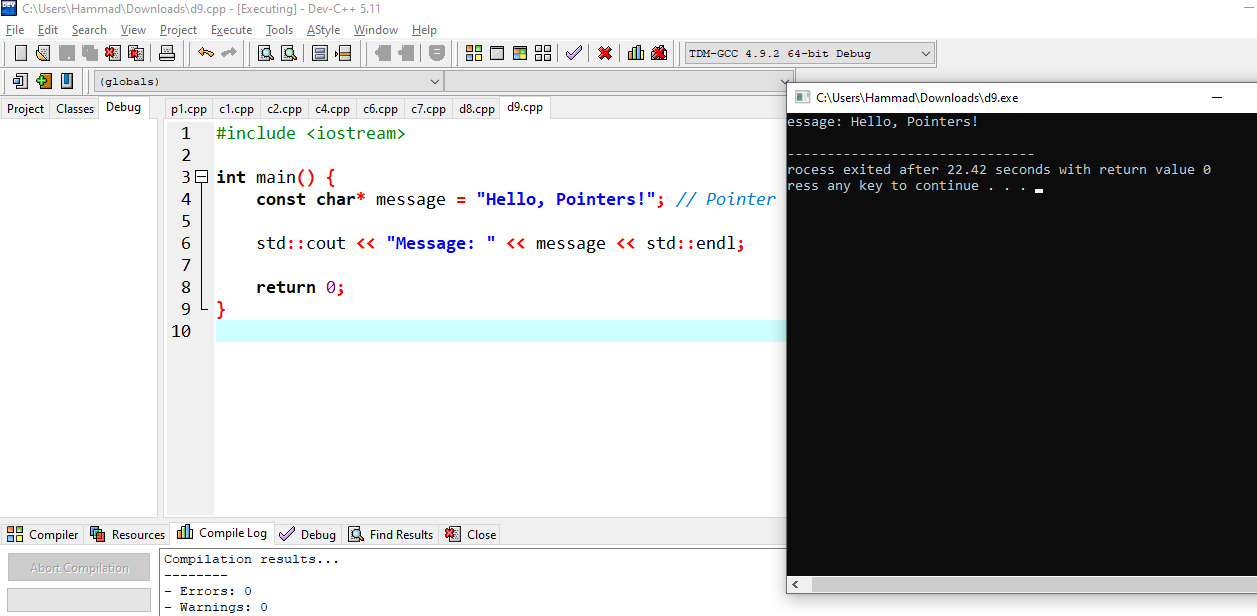
int main() {

const char\* message = "Hello, Pointers!"; // Pointer to a string literal

std::cout << "Message: " << message << std::endl;

return 0;

}



**Program# 9**

#include <iostream>

Using namespace std;

int main() {

int rows, cols;

std::cout << "Enter the number of rows and columns: ";

std::cin >> rows >> cols;

int\*\* matrix = new int\*[rows]; // Allocate an array of int pointers

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

matrix[i] = new int[cols]; // Allocate each row dynamically

}

// Initialize and print the 2D array

int count = 1;

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

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

matrix[i][j] = count++;

std::cout << matrix[i][j] << " ";

}

std::cout << std::endl;

}

// Deallocate memory

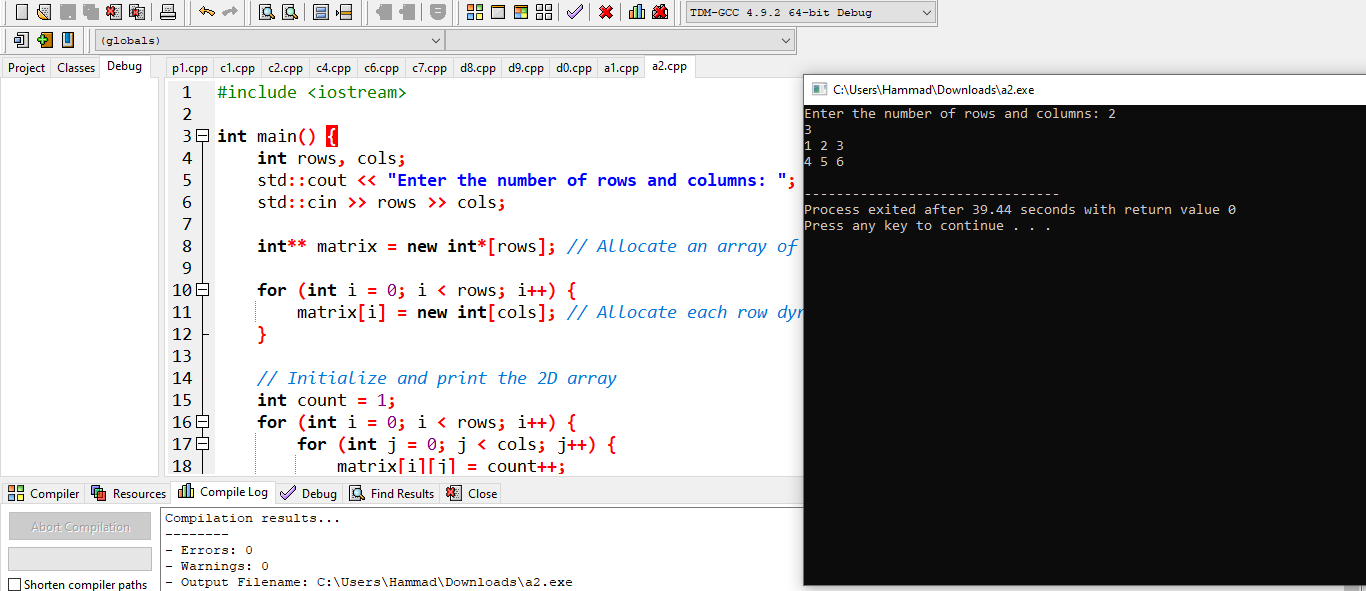
for (int i = 0; i < rows; i++) {

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

**Program # 10:**

#include <iostream>

Using namespace std;

void processNumbers(int\* numbers, int size, void (\*callback)(int)) {

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

callback(numbers[i]);

}

}

void printNumber(int num) {

std::cout << num << " ";

}

int main() {

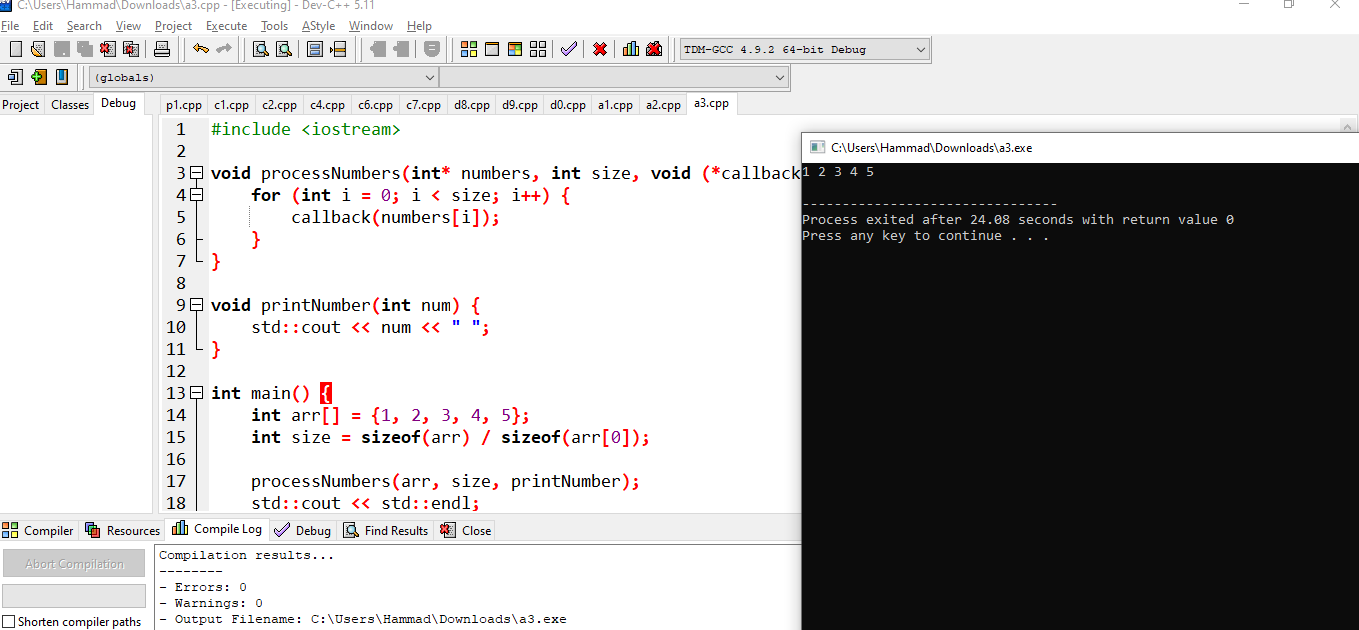
int arr[] = {1, 2, 3, 4, 5};

int size = sizeof(arr) / sizeof(arr[0]);

processNumbers(arr, size, printNumber);

std::cout << std::endl;

return 0;

}

**Program # 11:**

#include <iostream>

Using namespace std;

int main() {

const char\* message = "Hello, Pointers!"; // Pointer to a C-string

// Print each character of the string using pointer arithmetic

for (const char\* ptr = message; \*ptr != '\0'; ptr++) {

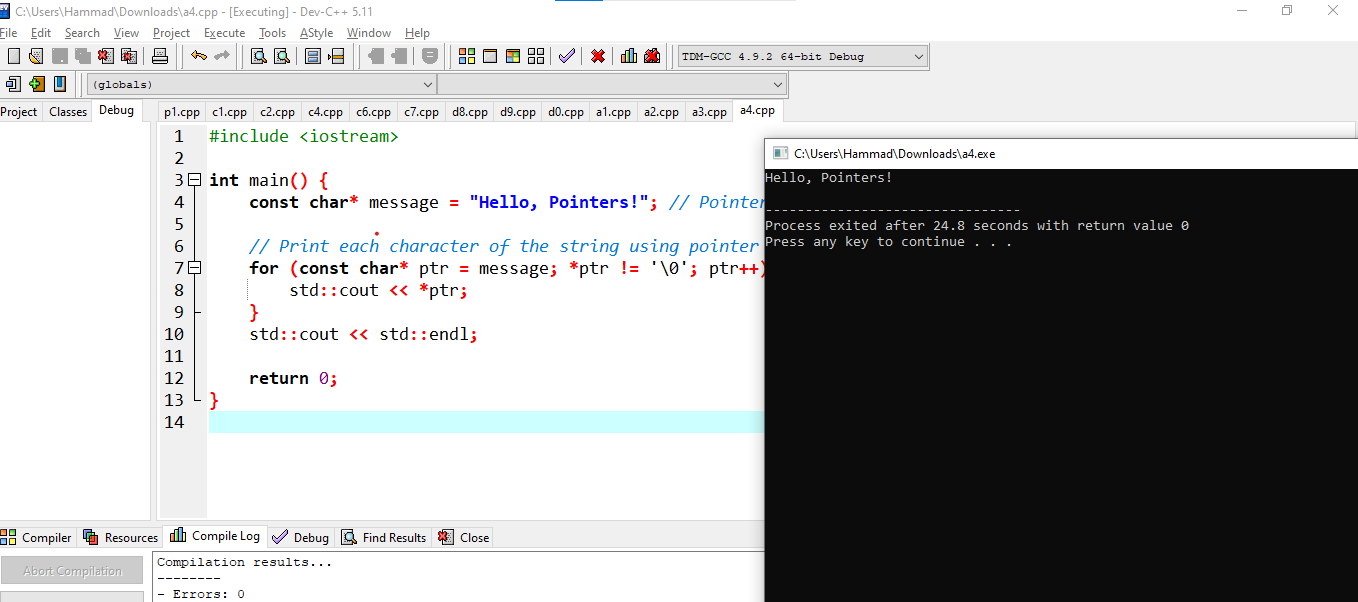
std::cout << \*ptr;

}

std::cout << std::endl;

return 0;

}



**Program # 12:**

#include <iostream>

#include <fstream>

int main() {

std::ofstream outfile("sample.txt");

if (!outfile) {

std::cerr << "Error opening file!" << std::endl;

return 1;

}

int data = 42;

int\* ptr = &data;

outfile << "Value: " << \*ptr << std::endl;

outfile.close();

std::ifstream infile("sample.txt");

std::string line;

if (infile) {

while (std::getline(infile, line)) {

std::cout << line << std::endl;

}

infile.close();

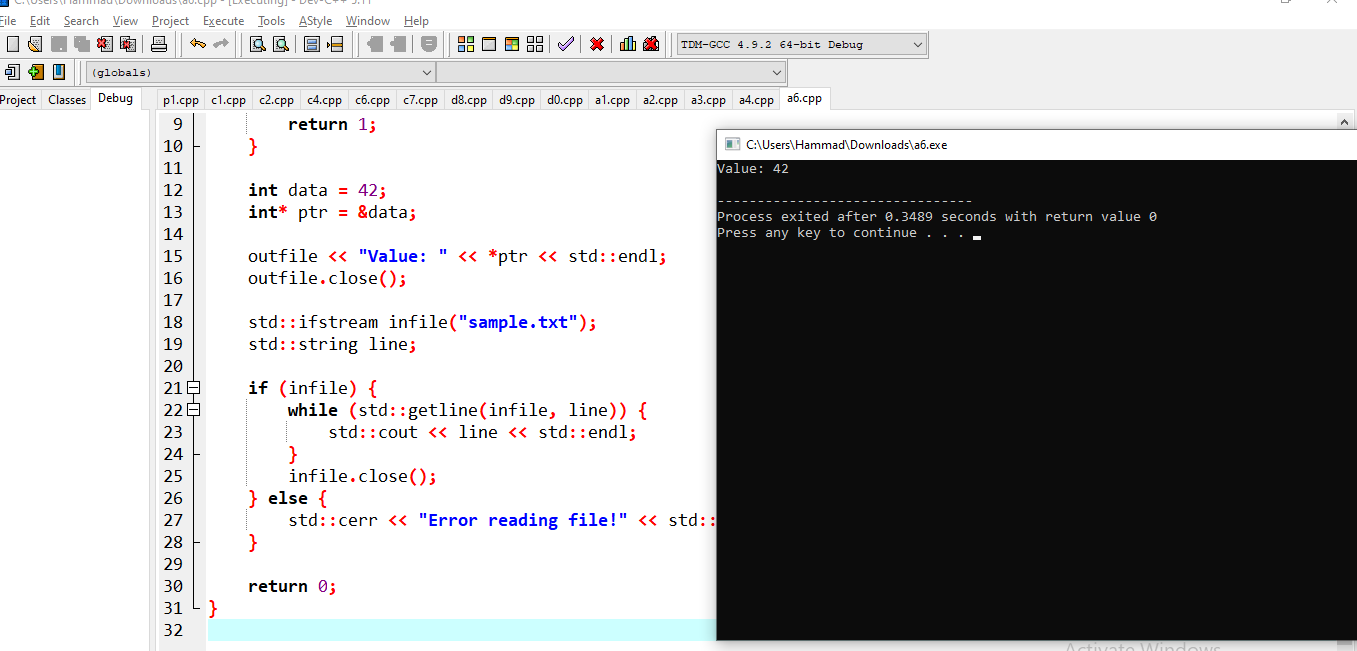
} else {

std::cerr << "Error reading file!" << std::endl;

}

return 0;

}



**Program #13:**

#include <iostream>

int main() {

int\* ptr = new int; // Allocate memory for an integer

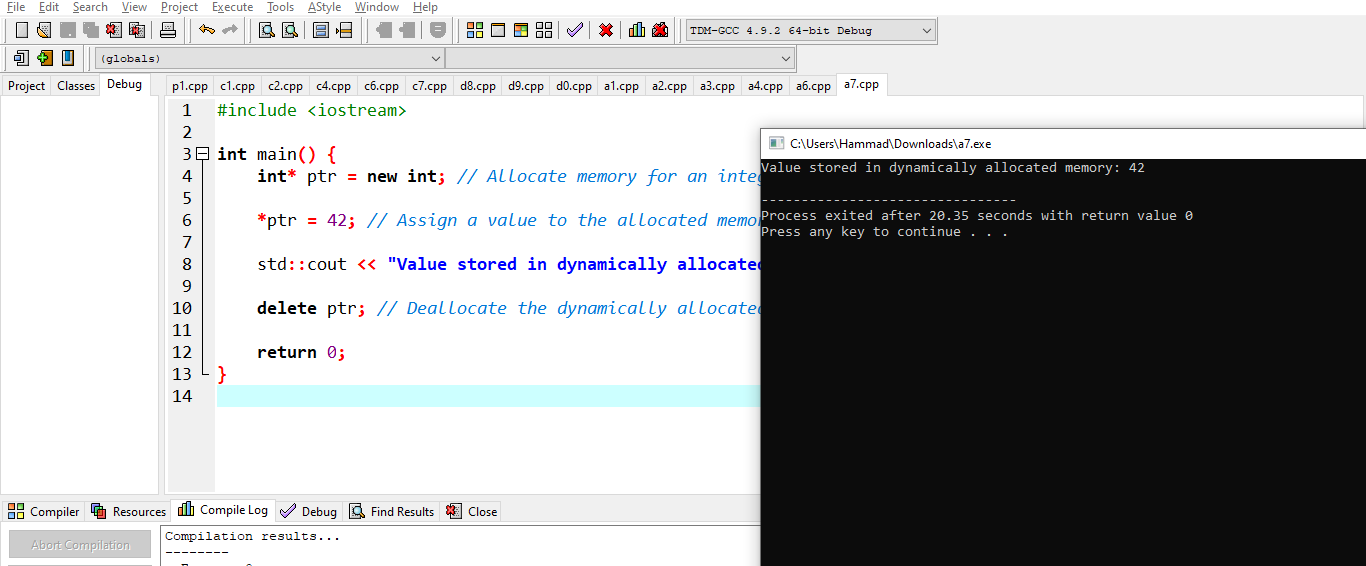
\*ptr = 42; // Assign a value to the allocated memory

std::cout << "Value stored in dynamically allocated memory: " << \*ptr << std::endl;

delete ptr; // Deallocate the dynamically allocated memory

return 0;

}



**Program # 14:**

#include <iostream>

void swap(int\* a, int\* b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

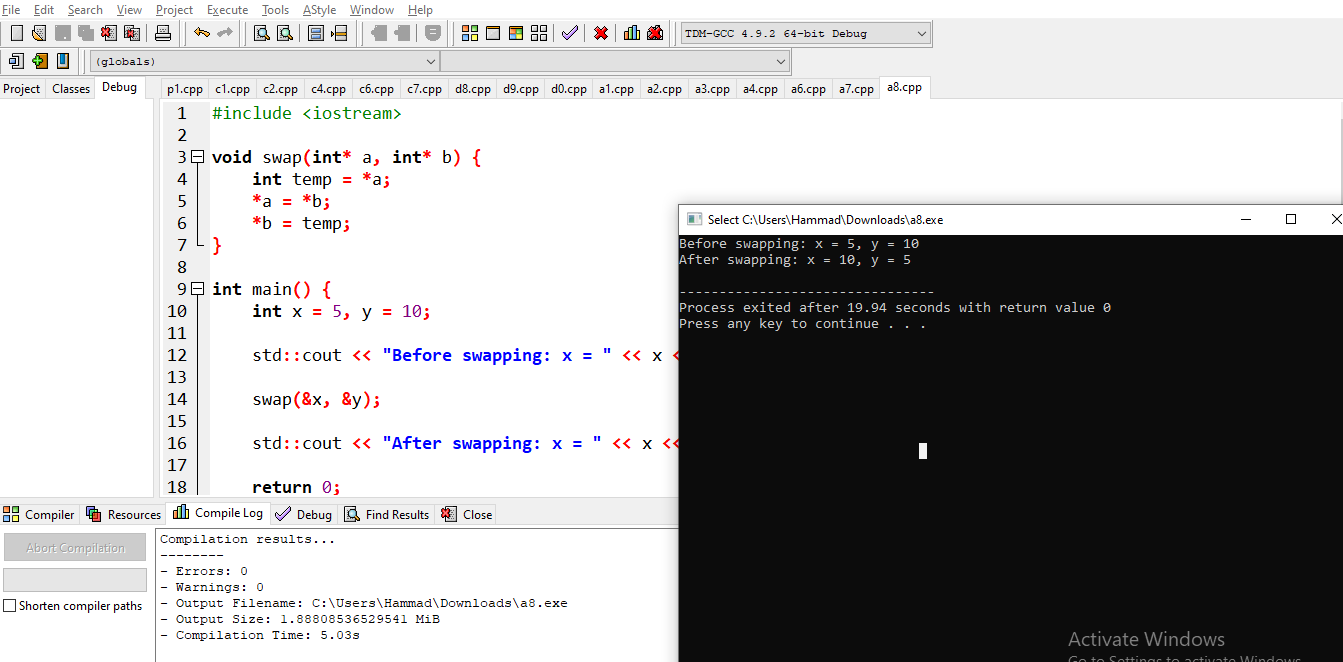
int x = 5, y = 10;

std::cout << "Before swapping: x = " << x << ", y = " << y << std::endl;

swap(&x, &y);

std::cout << "After swapping: x = " << x << ", y = " << y << std::endl;

return 0;

} 

**Program # 15:**

#include <iostream>

int main() {

int numbers[] = {7, 2, 9, 1, 5};

int size = sizeof(numbers) / sizeof(numbers[0]);

int max = numbers[0];

int\* ptr = numbers;

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

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

max = \*(ptr + i);

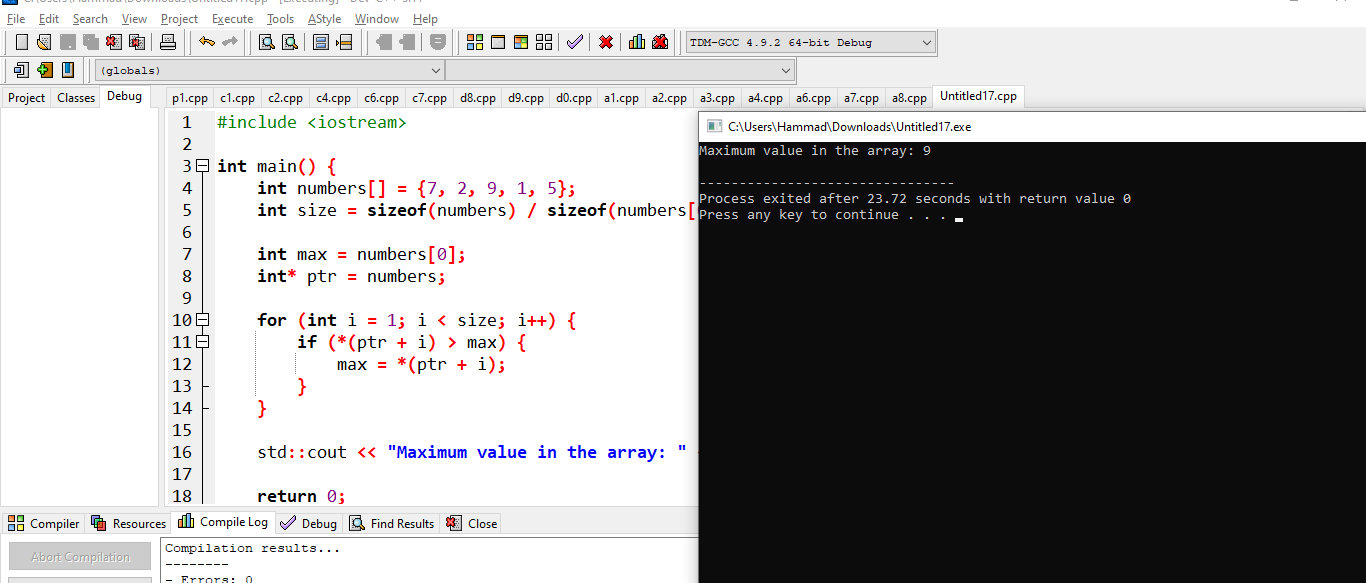
}

}

std::cout << "Maximum value in the array: " << max << std::endl;

return 0;

}



**Activity 2:**

How do I create my account on GitHub?

* IOpen my web browser and go to the GitHub website by entering the URL: <https://github.com/>.
* On the GitHub homepage, I see a "Sign up" button. So I click on it to start the registration process.
* Then I fill out the registration form with the following information:

**Username:** I choose a unique username (Anum-batool) for my GitHub account.

**Email address:** Enter a valid email address t.

**Password:** Then I Create a strong password for my GitHub account.

* GitHub ask me to complete a CAPTCHA challenge to verify that I am not a robot.
* GitHub send a verification email to the email address I provided. I click the verification link in the email to verify my email address.
* Once I've completed these steps, my GitHub account is created and verified.