

|  |  |
| --- | --- |
| Name: | Shaheen |
| Reg No: | Sp22-Bcs-071 |
| Submission Date: | 10/10/2023 |
| Submitted To: | Yasmeen Jana |

Comsat University Islamabad Vehari Campus

## Question 01: What a process to Create GitHub account?

1. Open a Web Browser:

Open your preferred web browser and go to the GitHub website at https://github.com.

2. Sign UP:

On the GitHub homepage, you will see a "Sign up" button in the upper right corner. Click on it.

3. Create GitHub Account:

You will be directed to a sign-up page. Here, you will need to provide the following information:

Username:

Choose a unique username for your GitHub account. This will also be a part of your GitHub profile URL (e.g., https://github.com/your-username).

Email Address:

Enter your email address. This will be used for account notifications and password resets.

Password:

Create a strong password for your GitHub account.

4. Verify Your Email Address:

After you've filled in the required information, GitHub will send you a verification email to the address you provided. Open your email inbox and click the verification link in the email to confirm your email address.

5. Choose a Plan:

GitHub offers both free and paid plans. For most users, the free plan (GitHub Free) is sufficient. Click on the "Choose Free" button to continue.

6. Welcome to GitHub:

Once you are completed all the steps, you'll be directed to your GitHub dashboard, and you'll officially have a GitHub account.

Question 02: Write 15 programs that will explain the concepts of pointer.

Program 1: Declaring a pointer and accessing a variable.

#include <iostream>

using namespace std;

int main() {

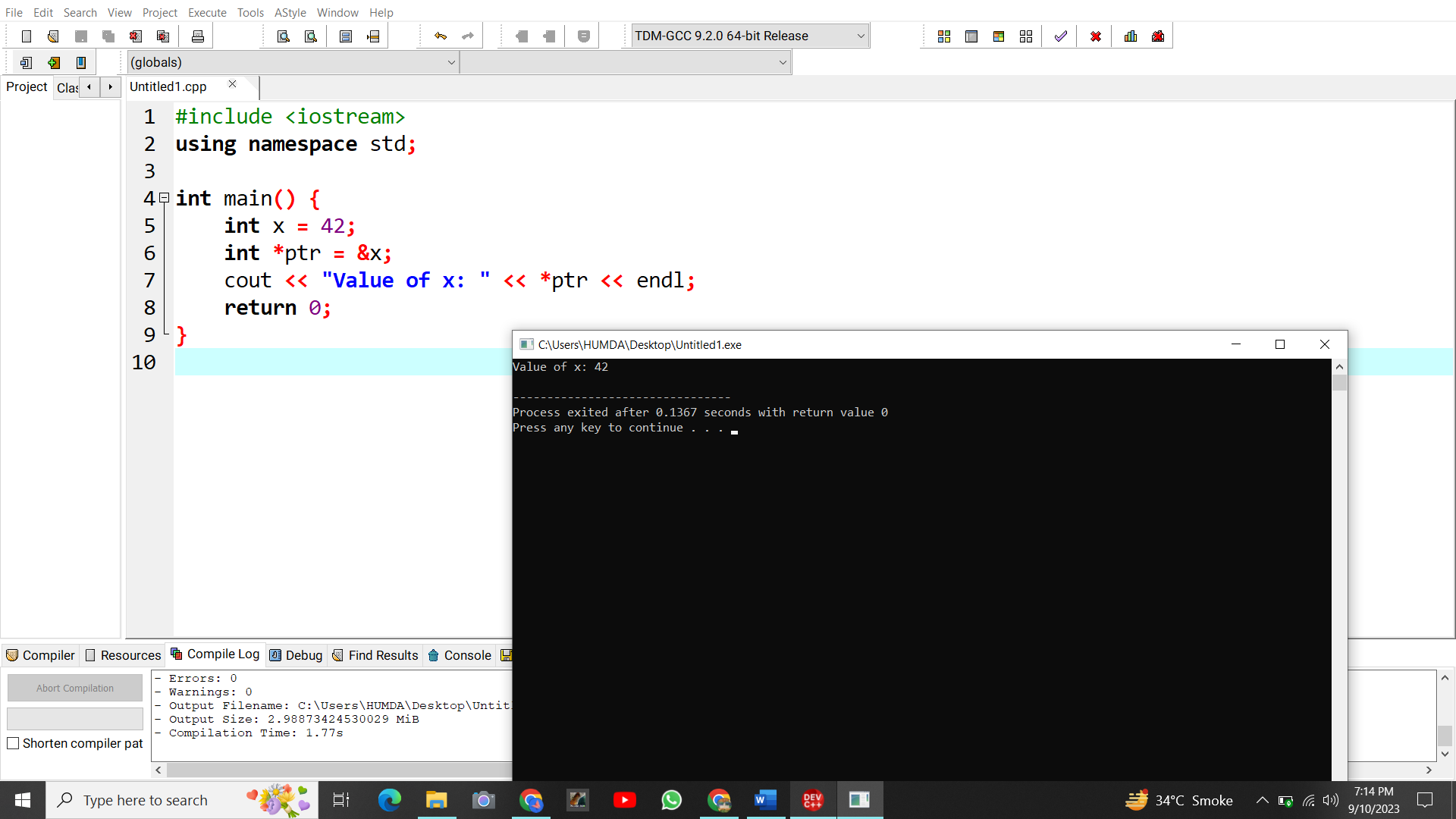
int x = 42;

int \*ptr = &x;

cout << "Value of x: " << \*ptr << endl;

return 0;

}



Program 2: Modifying a variable through a pointer.

#include <iostream>

using namespace std;

int main() {

int x = 42;

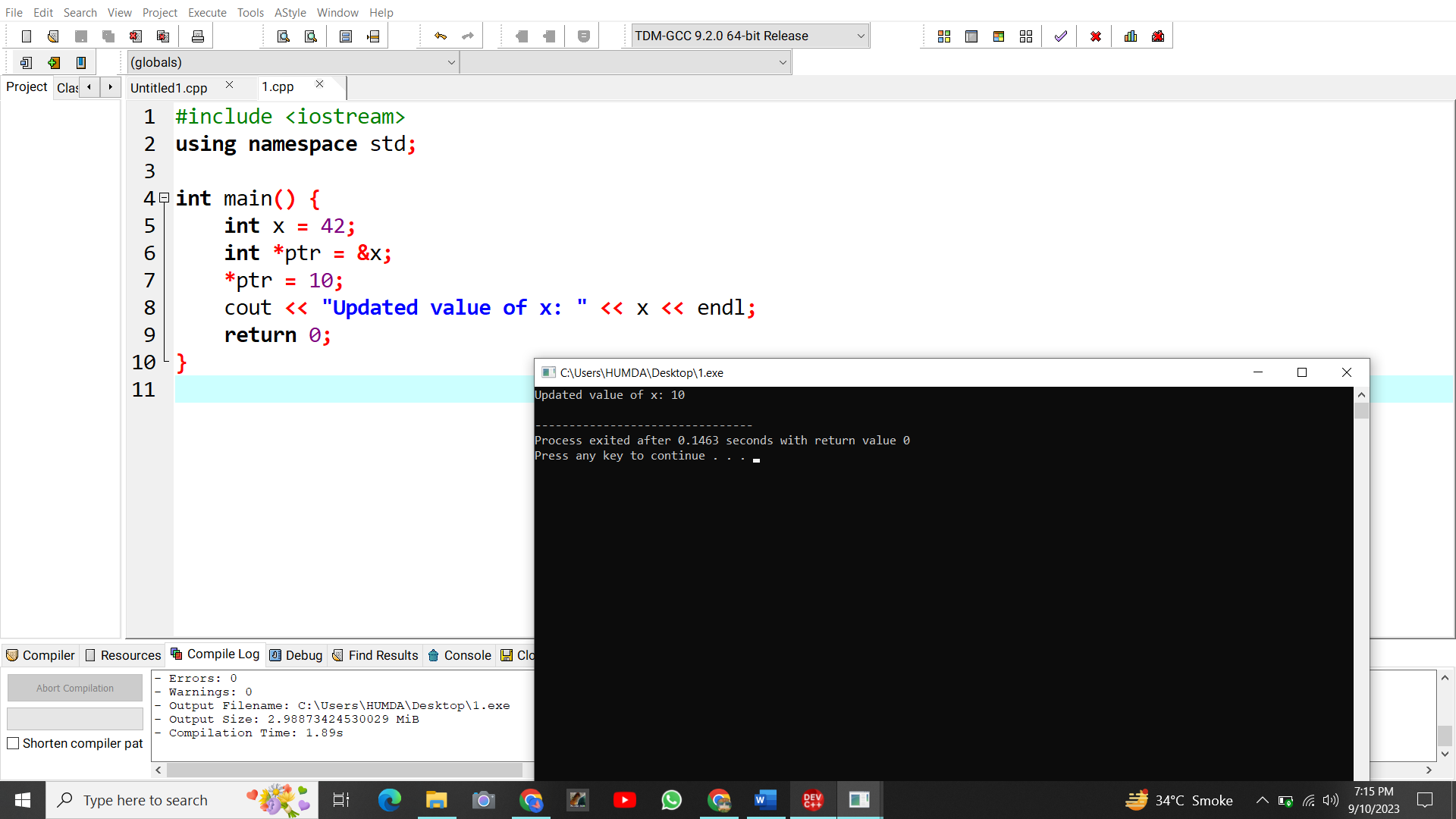
int \*ptr = &x;

\*ptr = 10;

cout << "Updated value of x: " << x << endl;

return 0;

}



Program 3: Null pointer.

#include <iostream>

using namespace std;

int main() {

int \*nullPtr = nullptr;

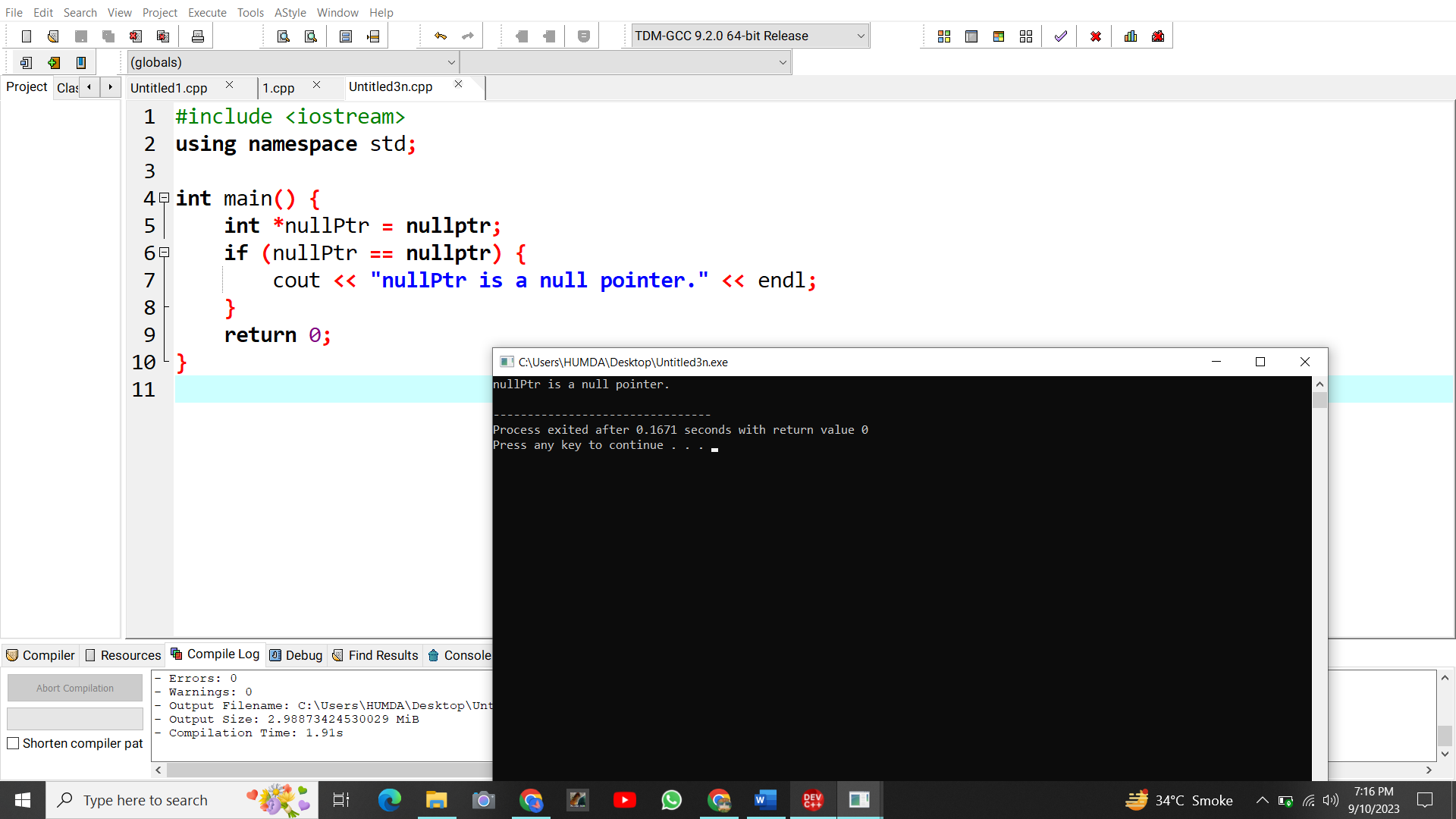
if (nullPtr == nullptr) {

cout << "nullPtr is a null pointer." << endl;

}

return 0;

}



Program 4: Pointer arithmetic.

#include <iostream>

using namespace std;

int main() {

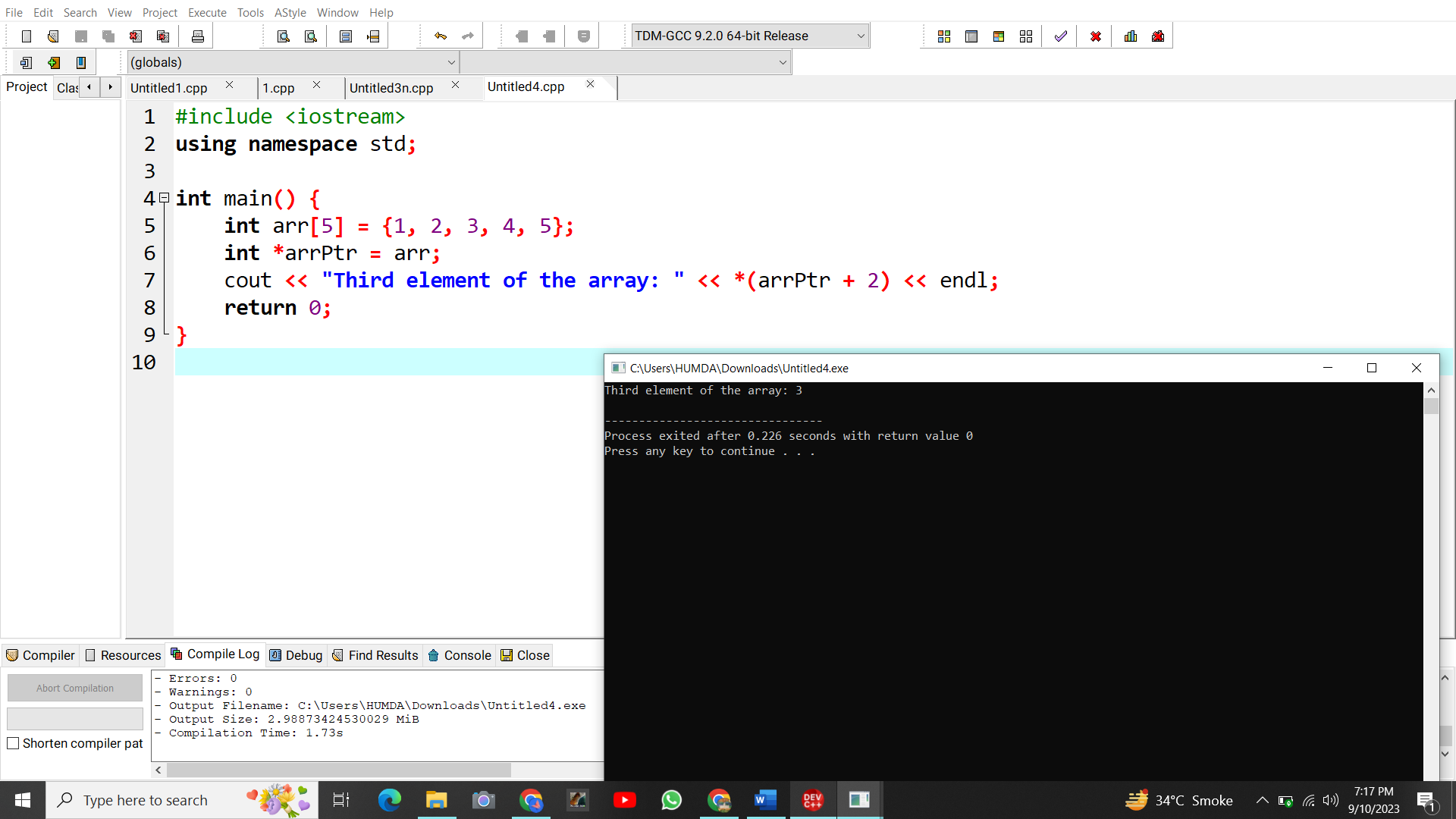
int arr[5] = {1, 2, 3, 4, 5};

int \*arrPtr = arr;

cout << "Third element of the array: " << \*(arrPtr + 2) << endl;

return 0;

}



Program5: Swapping two variables using pointers.

#include <iostream>

using namespace std;

int main() {

int a = 5, b = 10;

int \*ptrA = &a, \*ptrB = &b;

int temp = \*ptrA;

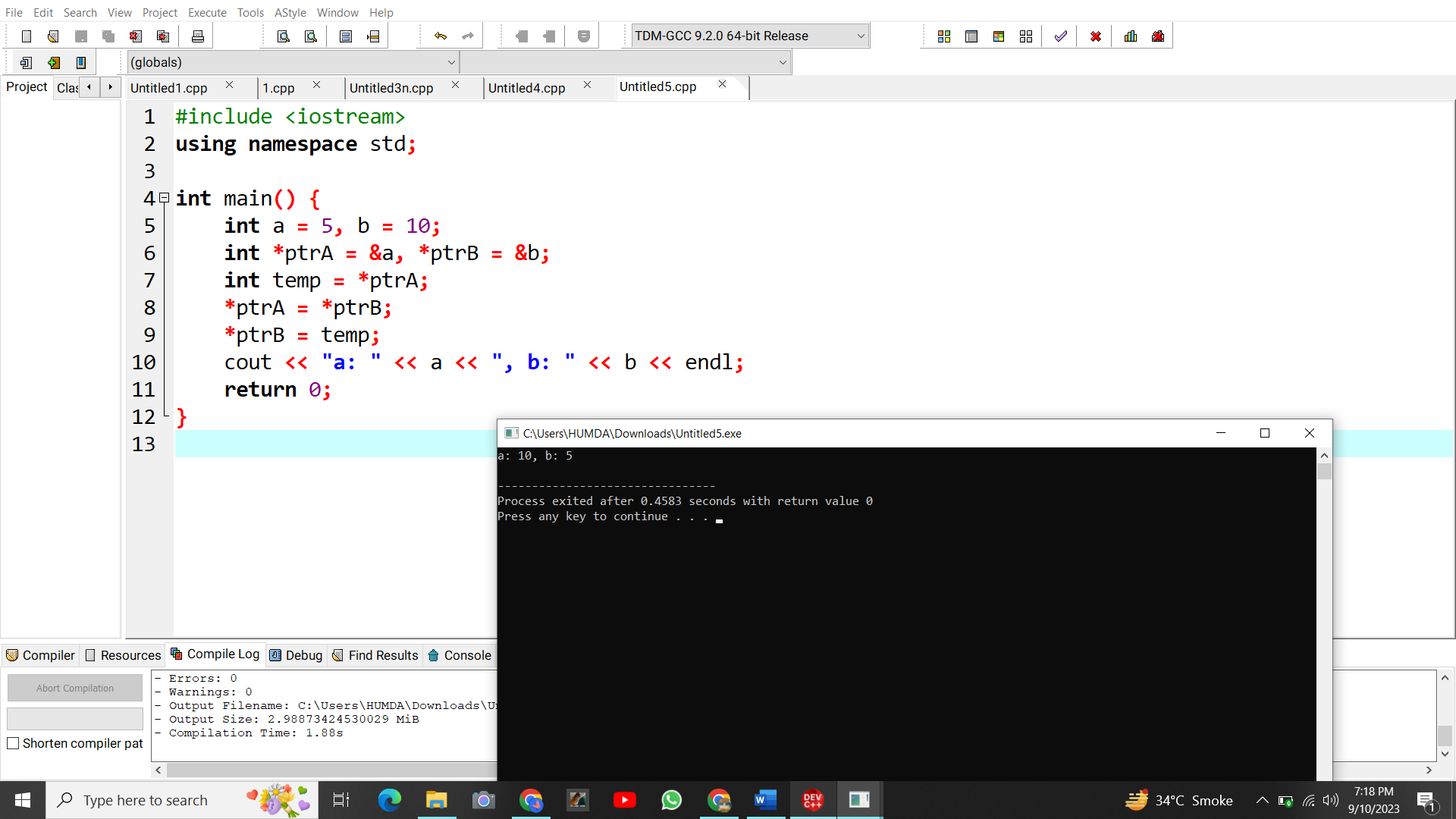
\*ptrA = \*ptrB;

\*ptrB = temp;

cout << "a: " << a << ", b: " << b << endl;

return 0;

}



Program 6: Dynamic memory allocation.

#include <iostream>

using namespace std;

int main() {

int \*dynPtr = new int;

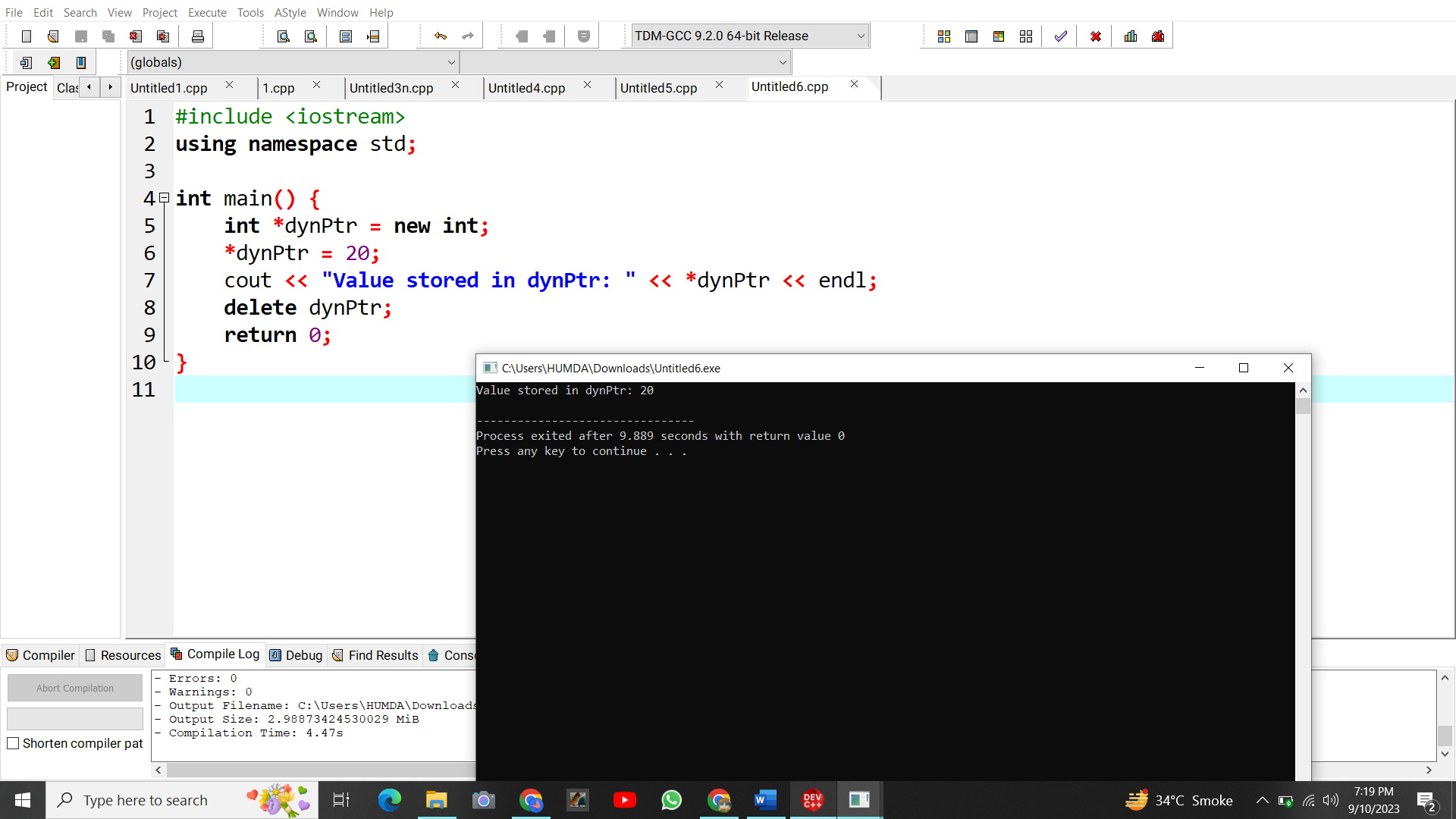
\*dynPtr = 20;

cout << "Value stored in dynPtr: " << \*dynPtr << endl;

delete dynPtr;

return 0;

}



Program 7: Array of pointers.

#include <iostream>

using namespace std;

int main() {

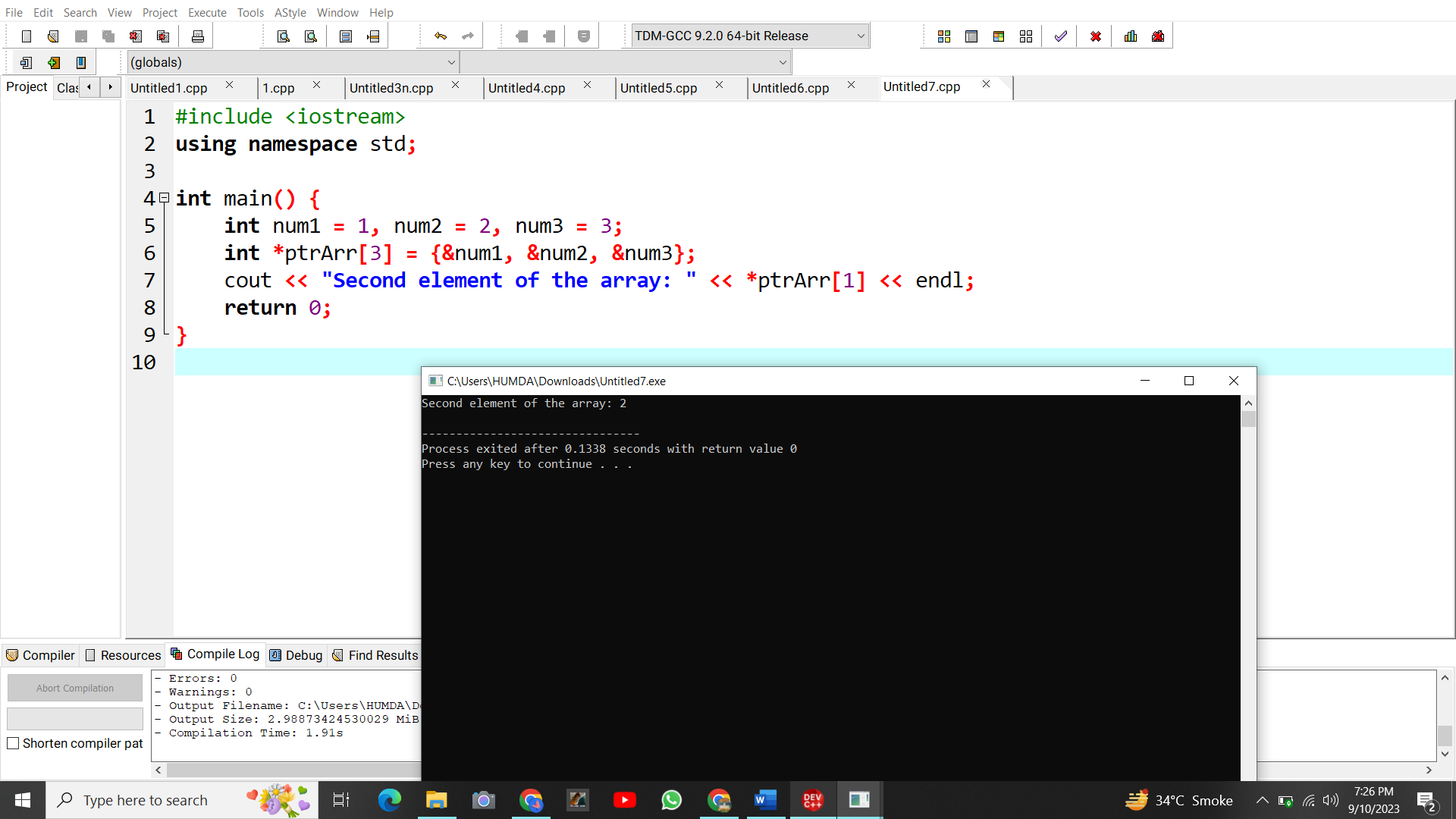
int num1 = 1, num2 = 2, num3 = 3;

int \*ptrArr[3] = {&num1, &num2, &num3};

cout << "Second element of the array: " << \*ptrArr[1] << endl;

return 0;

}



Program 8: Pointer to Pointer.

#include <iostream>

using namespace std;

int main() {

int y = 100;

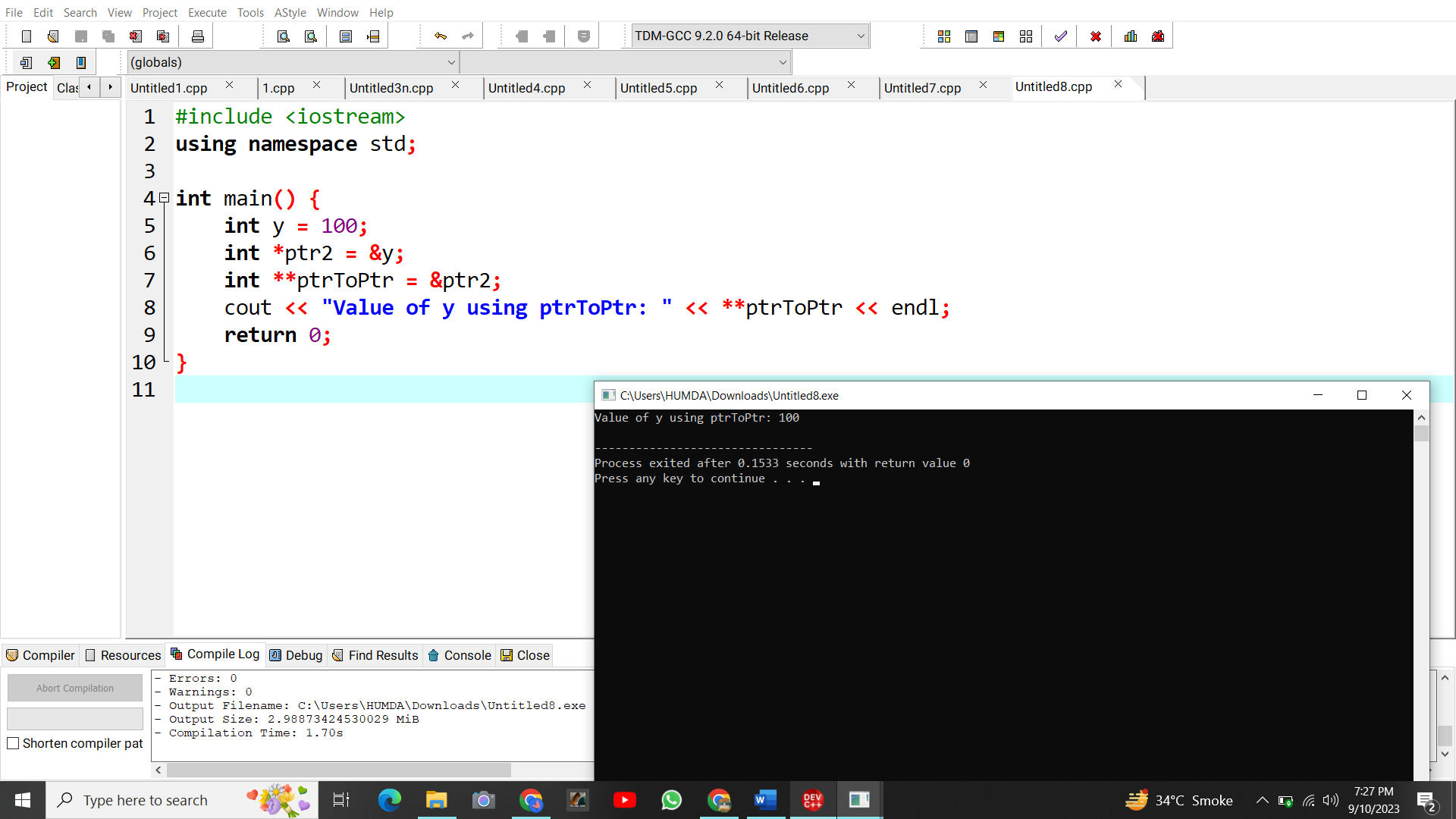
int \*ptr2 = &y;

int \*\*ptrToPtr = &ptr2;

cout << "Value of y using ptrToPtr: " << \*\*ptrToPtr << endl;

return 0;

}



Program 9: Const pointer.

#include <iostream>

using namespace std;

int main() {

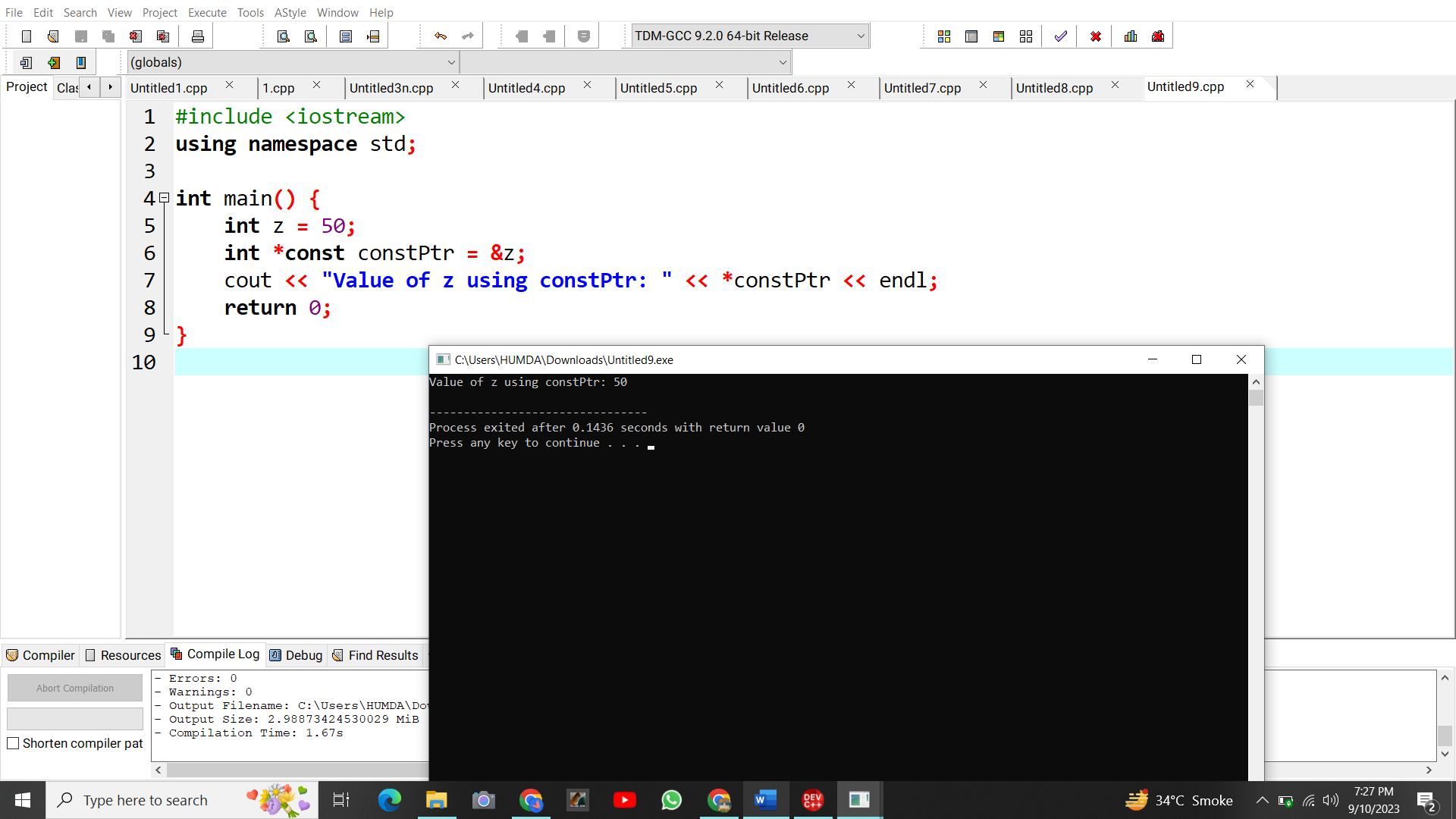
int z = 50;

int \*const constPtr = &z;

cout << "Value of z using constPtr: " << \*constPtr << endl;

return 0;

}



Program 10: Passing pointers to functions.

#include <iostream>

using namespace std;

void modifyValue(int \*ptr) {

\*ptr = 40;

}

int main() {

int value = 30;

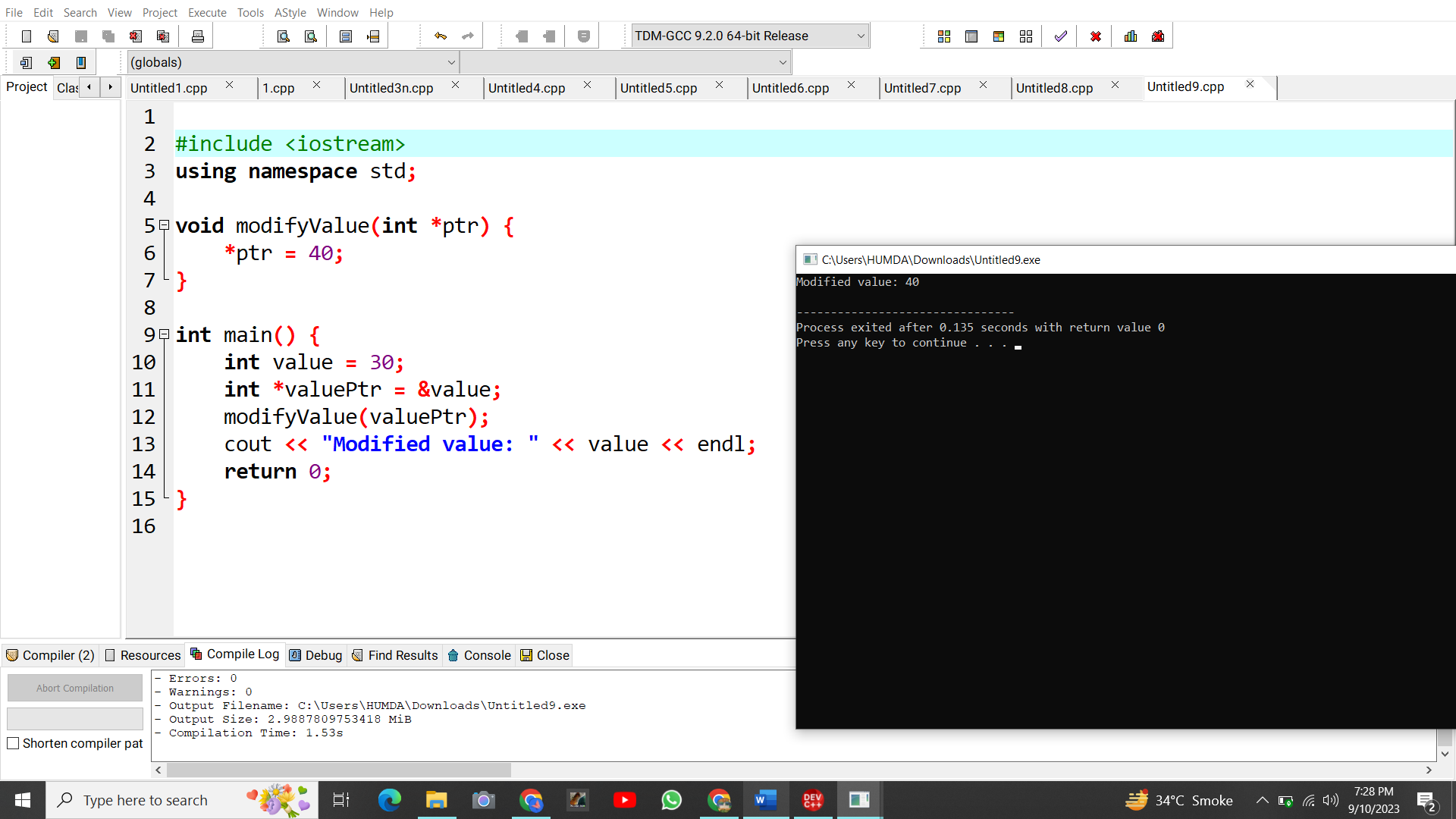
int \*valuePtr = &value;

modifyValue(valuePtr);

cout << "Modified value: " << value << endl;

return 0;

}



Program 11: void pointer.

#include <iostream>

using namespace std;

int main() {

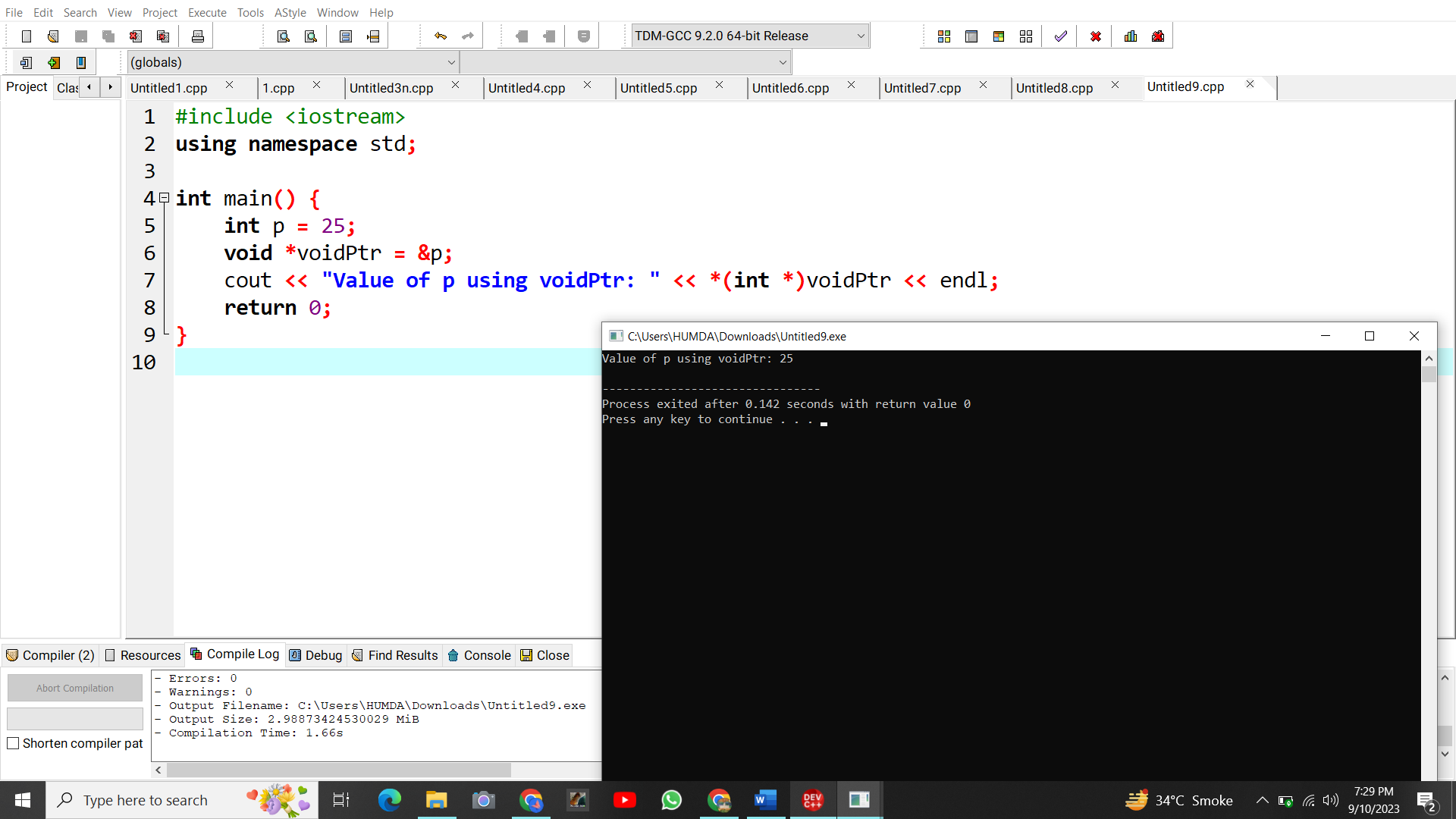
int p = 25;

void \*voidPtr = &p;

cout << "Value of p using voidPtr: " << \*(int \*)voidPtr << endl;

return 0;

}



Program 12: Pointer to a function.

#include <iostream>

using namespace std;

int add(int a, int b) {

return a + b;

}

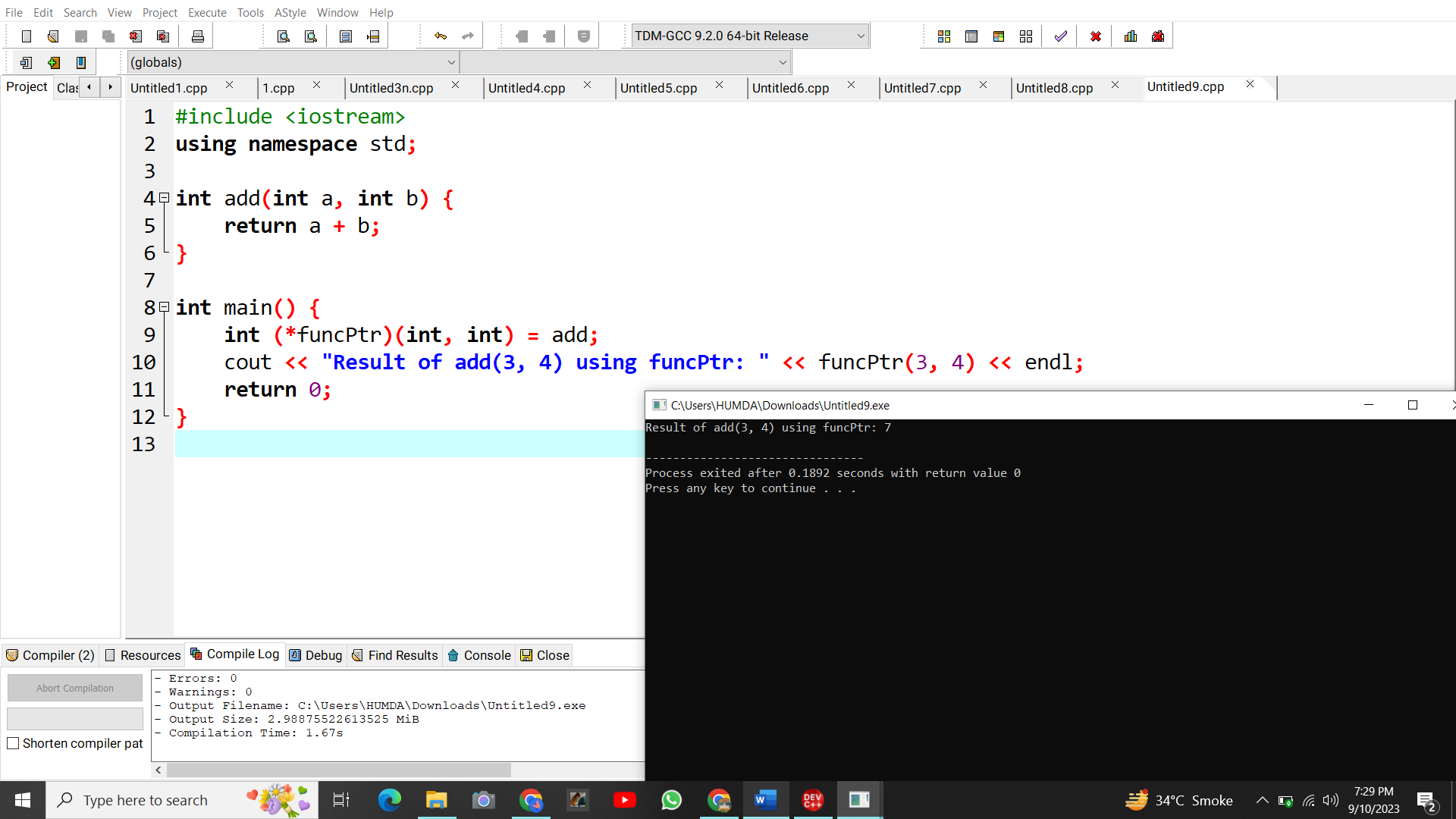
int main() {

int (\*funcPtr)(int, int) = add;

cout << "Result of add(3, 4) using funcPtr: " << funcPtr(3, 4) << endl;

return 0;

}



Program 13: Pointer member to variable.

#include <iostream>

using namespace std;

struct MyStruct {

int data;

};

int main() {

MyStruct obj;

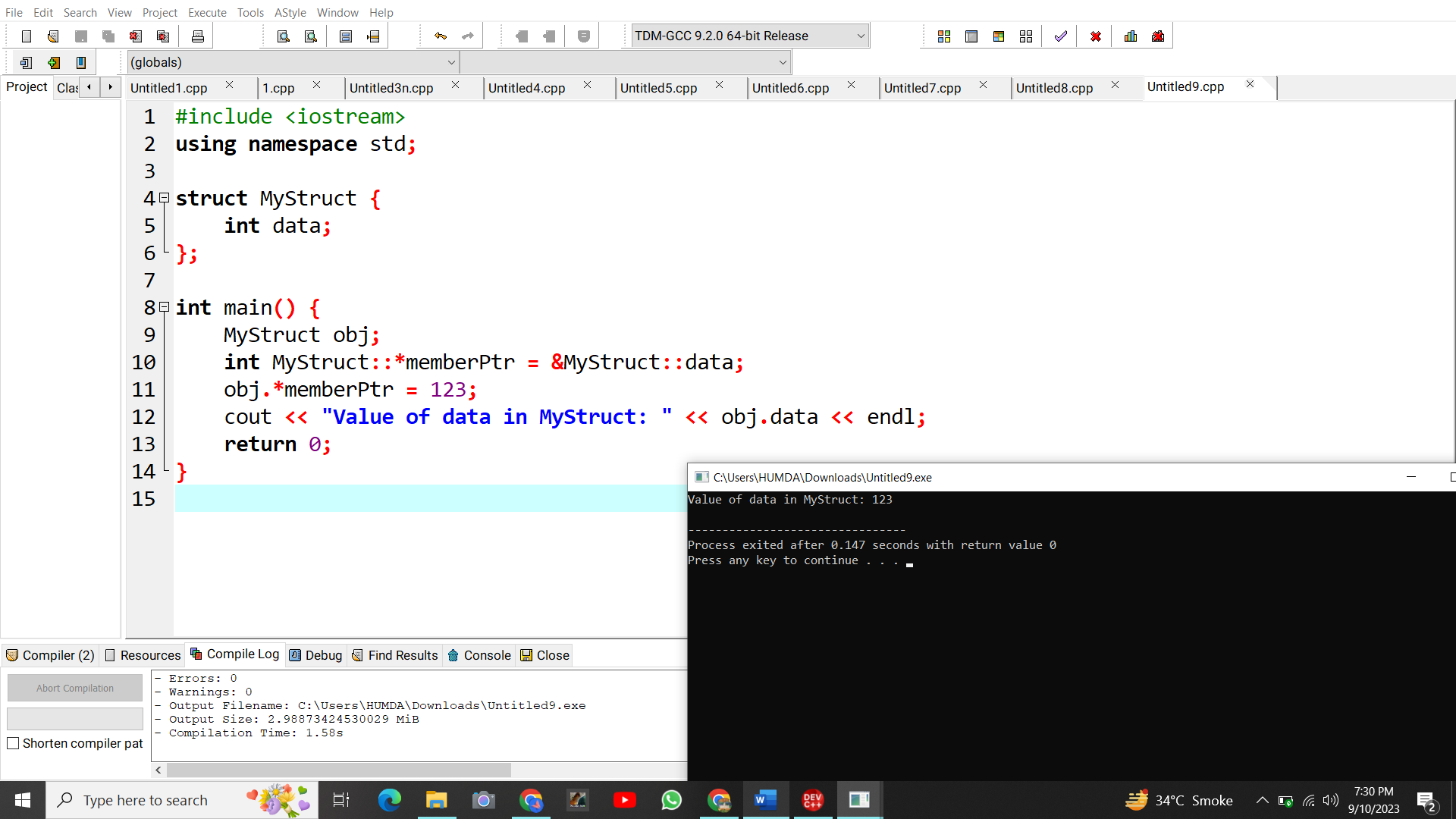
int MyStruct::\*memberPtr = &MyStruct::data;

obj.\*memberPtr = 123;

cout << "Value of data in MyStruct: " << obj.data << endl;

return 0;

}



Program 14:Pointer to member function.

#include <iostream>

using namespace std;

struct MyStruct {

int data;

int getData() {

return data;

}

};

int main() {

MyStruct obj;

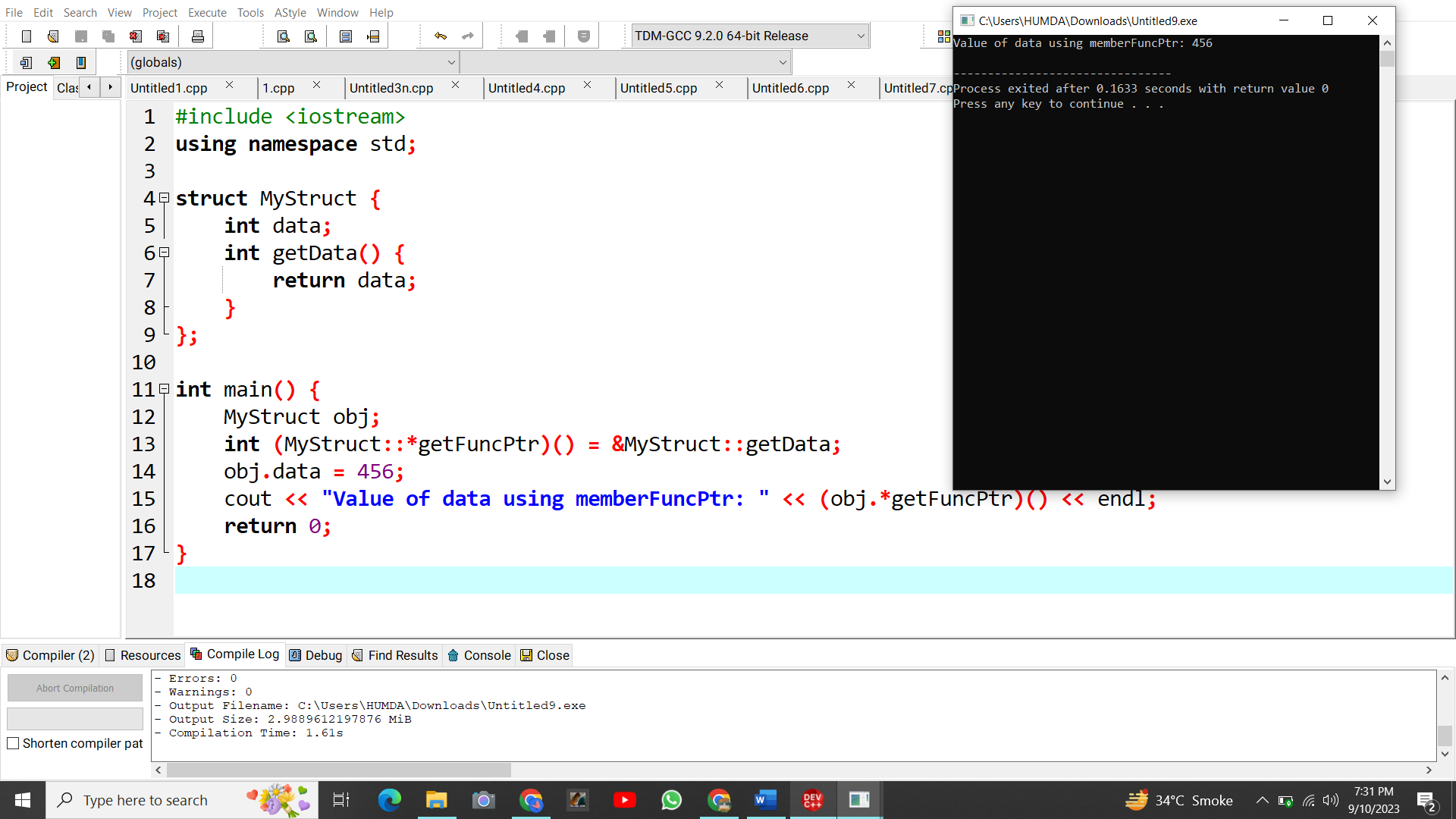
int (MyStruct::\*getFuncPtr)() = &MyStruct::getData;

obj.data = 456;

cout << "Value of data using memberFuncPtr: " << (obj.\*getFuncPtr)() << endl;

return 0;

}



Program 15:Add two number using pointer.

#include <iostream>

using namespace std;

int main() {

int num1, num2;

cout << "Enter the first number: ";

cin >> num1;

cout << "Enter the second number: ";

cin >> num2;

int sum = num1 + num2;

int \*resultPtr = &sum;

cout << "Sum of " << num1 << " and " << num2 << " is: " << \*resultPtr << endl;

return 0;

}

