

LAB Task

Submitted to:

Mam Yasmeeen Jana

Submitted by:

Hassan Raza Bhatti

Registration No:

SP22-BCS-096

Section:

"C"

Subject:

Data Structure and algorithm

Date:

10-sep-2023



**Comsats University Islamabad,
Vehari campus.**

Activity 1:

Following are the creation steps:

Creating a GitHub account is a straightforward process that anyone can follow. Here's a step-by-step guide on how I created my GitHub account:

Step 1: Visit the GitHub Website:

Open your web browser and visit the GitHub website: [\[https://github.com\]](https://github.com).

Step 2: Sign Up:

Click on the "Sign Up" button in the top-right corner of the GitHub homepage.

Step 3: Provide Your Username, Email, and Password:

You'll be directed to the Sign Up form. Here, you need to provide the following information:

Username: Choose a username that will be your GitHub handle. It should be easy to remember as it will be a part of your GitHub profile URL.

Email address: Enter a valid email address that you have access to. GitHub will use this email for communication and account recovery.

Password: Create a strong password. GitHub will indicate the password's strength, so make sure it's secure.

Step 4: Complete the CAPTCHA:

GitHub asks you to complete a CAPTCHA to verify that you're not a robot. Follow the on-screen instructions to complete it.

Step 5: Choose Your Plan (Optional):

GitHub offers a choice between Free and paid plans. For most users, the free plan is sufficient. If you want to explore the features of the paid plans, you can do so later.

Step 6: Verify Your Email Address:

GitHub will send a verification email to the address you provided. Open your email inbox and click on the verification link within the email. This step is crucial to activate your GitHub account.

Step 7: Complete Your Profile:

After verifying your email, you can personalize your GitHub profile. Add a profile picture, write a short bio, and customize your profile to your liking.

Step 8: Explore GitHub:

Now that your account is set up, take some time to explore GitHub's interface. You can search for projects and repositories and follow other users or projects that interest you.

Congratulations! You've successfully created a GitHub account.

Activity 2:

Program 1:

```
#include <iostream>
```

```
Using namespace std;
```

```
int main () {
```

```
    int num = 10;
```

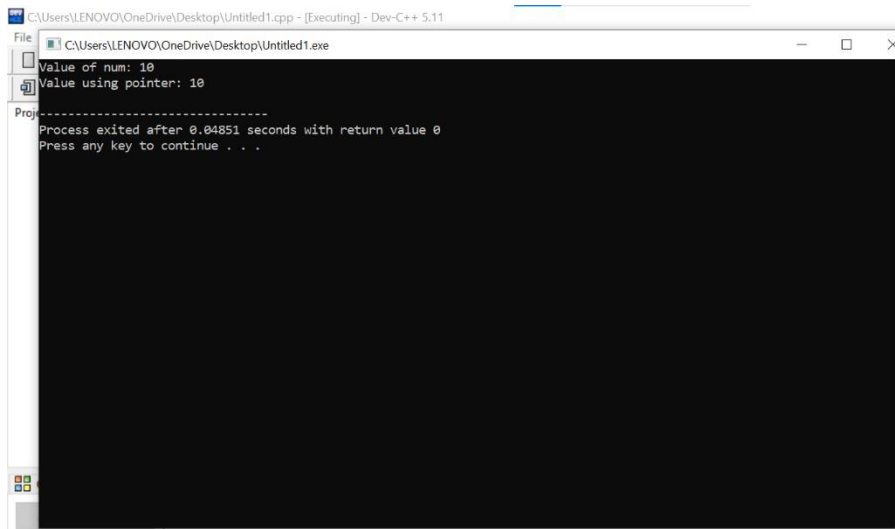
```
    int *ptr = &num;
```

```
    std::cout << "Value of num: " << num << std::endl;
```

```
    std::cout << "Value using pointer: " << *ptr << std::endl;
```

```
    return 0;
```

```
}
```



Program 2:

```
#include <iostream>
```

```
Using namespace std;
```

```

int main() {

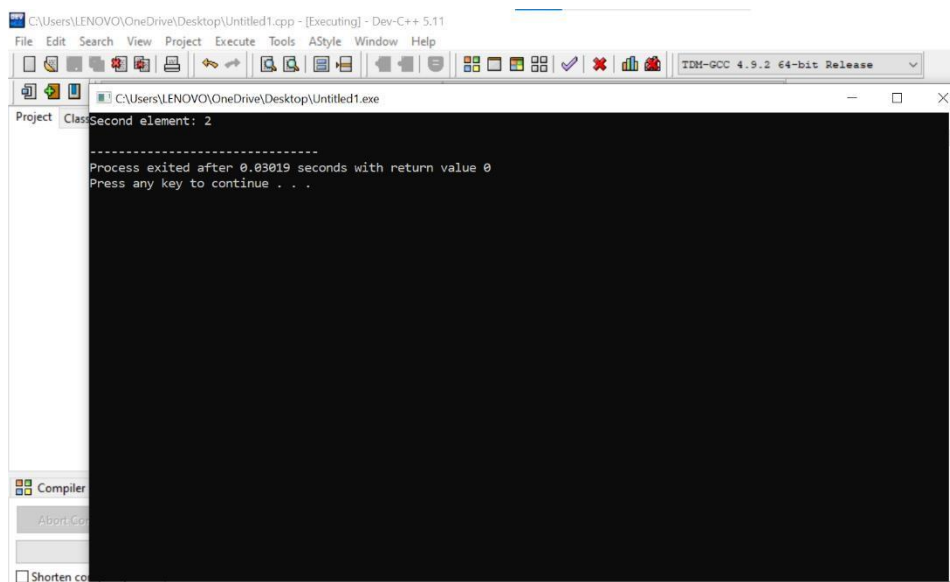
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;

    std::cout << "Second element: " << *(ptr + 1) <<
    std::endl;

    return 0;

}

```



Program 3;

```
#include <iostream>
```

Using namespace std;

```

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;
    for (int i = 0; i < 5; ++i) {
        std::cout << "Element " << i << ": " << *(ptr + i) <<
        std::endl;
    }
}

```

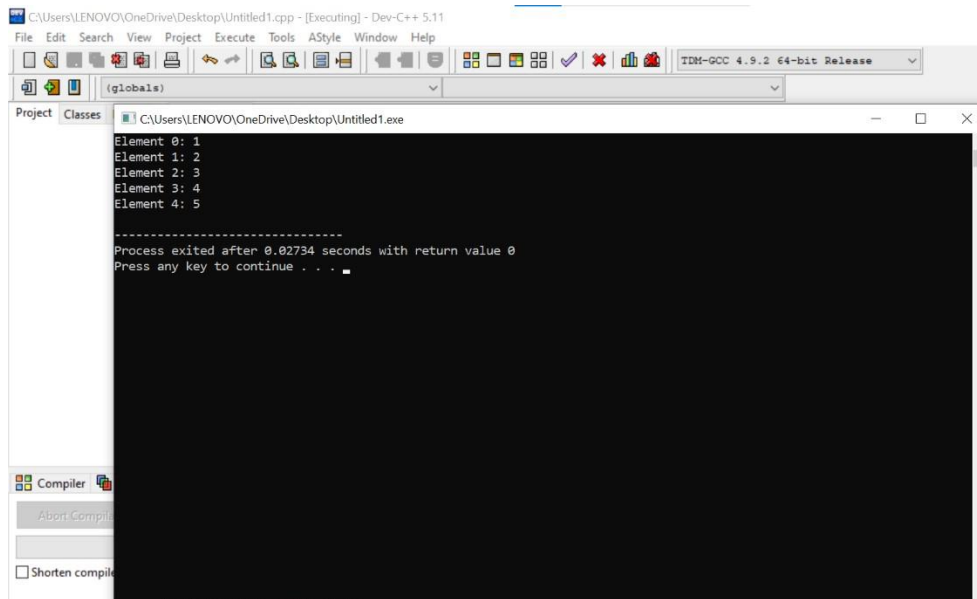
```

}

return 0;

}

```



Program 4:

```
#include <iostream>
```

```

void sayHello() { std::cout << "Hello,
World!" << std::endl;
}

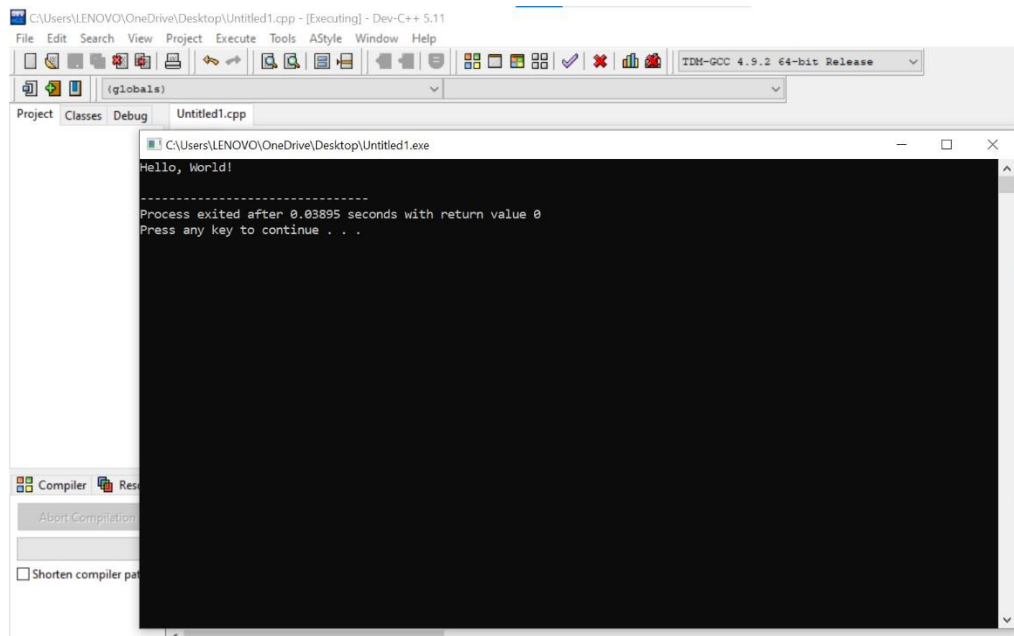
```

```

int main() {      void
(*func_ptr)() = &sayHello;

    (*func_ptr)(); // Call the function using a pointer
return 0;
}

```



Program 5:

```
#include <iostream>
```

```
struct Point {
```

```
    int x, y;
```

```
};
```

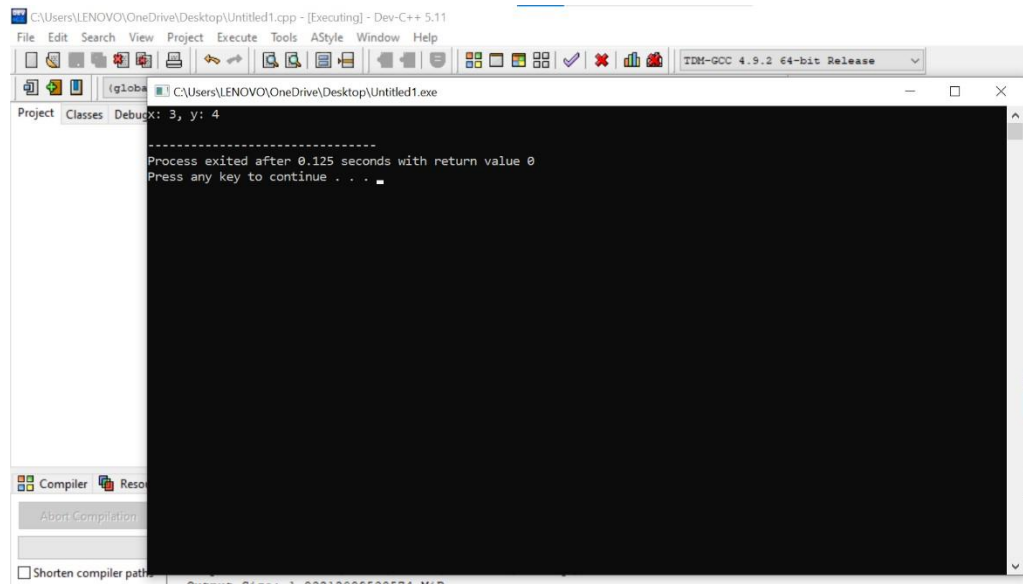
```
int main() {
```

```
    Point p = {3, 4};   Point *ptr = &p;   std::cout << "x:
```

```
" << ptr->x << ", y: " << ptr->y << std::endl;   return
```

```
0;
```

```
}
```

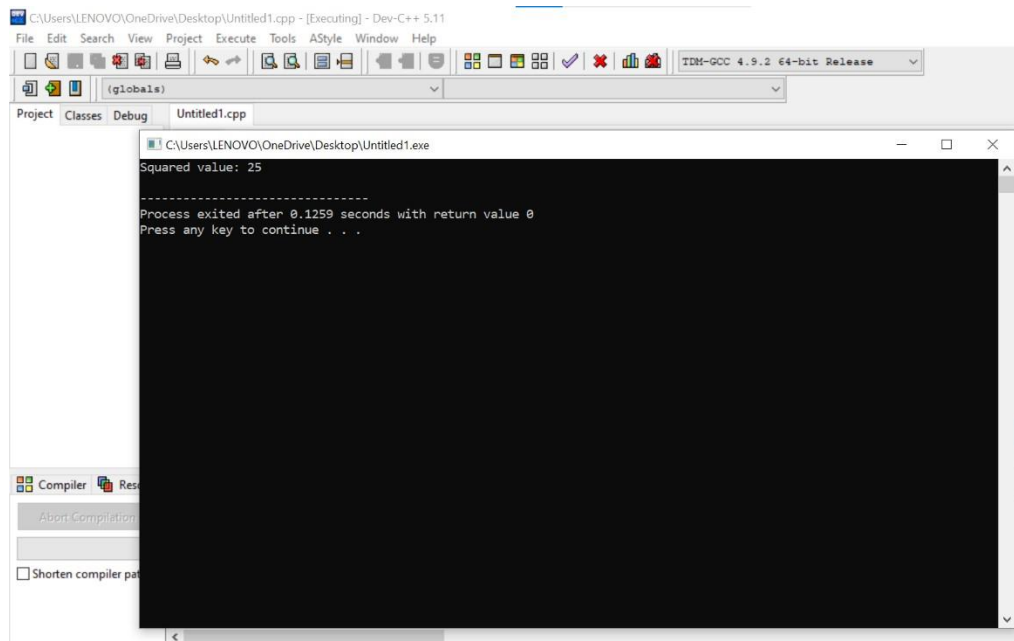


Program 6:

```
#include <iostream>
```

```
void square(int *ptr) {  
    *ptr = (*ptr) * (*ptr);  
}
```

```
int main() {    int num = 5;    square(&num);  
    std::cout << "Squared value: " << num <<  
    std::endl;    return 0;  
}
```

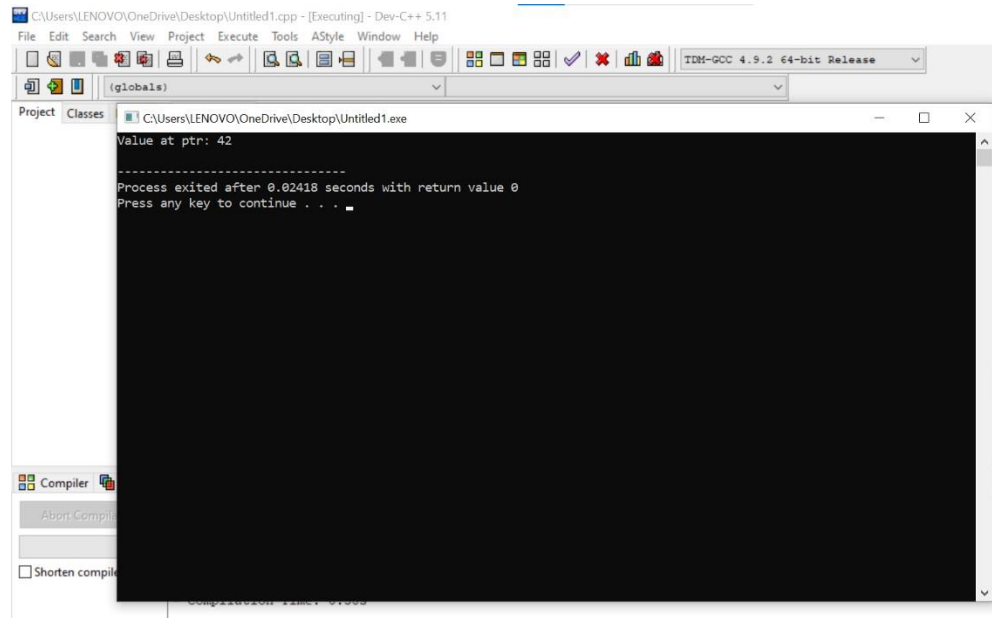



Program 7:

#include <iostream>

```
int main() { int *ptr = new int;  *ptr = 42;
std::cout << "Value at ptr: " << *ptr <<
std::endl;      delete ptr; // Deallocate
memory

}
```



Program 8:

```
#include <iostream>
```

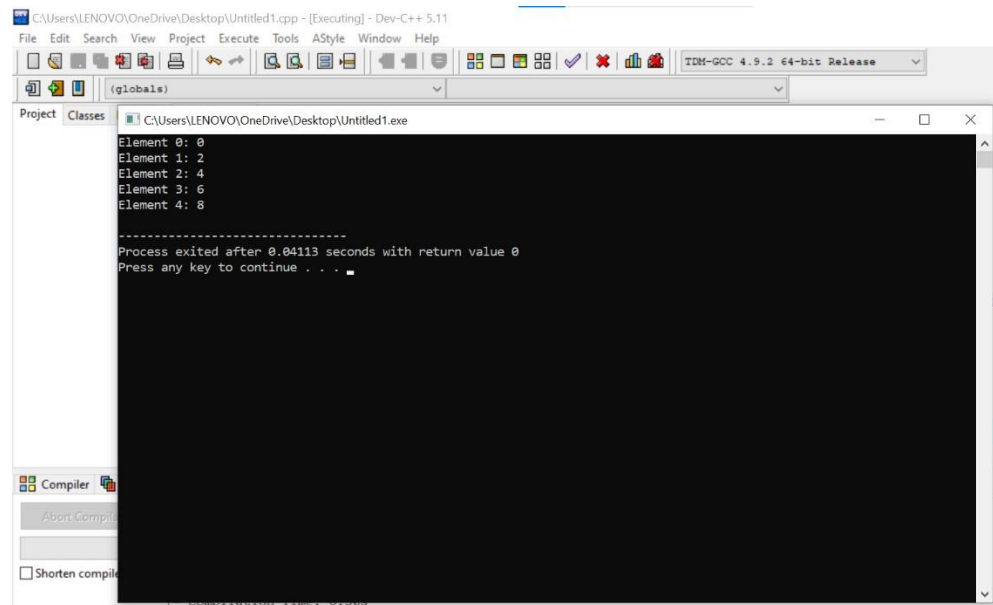
```
int main() {    int
*arr = new int[5];
    for (int i = 0; i < 5; ++i) {
arr[i] = i * 2;
    }

    for (int i = 0; i < 5; ++i) {    std::cout << "Element
" << i << ": " << arr[i] << std::endl;
    }

    delete[] arr; // Deallocate memory

    return 0;
```

}



Program 9:

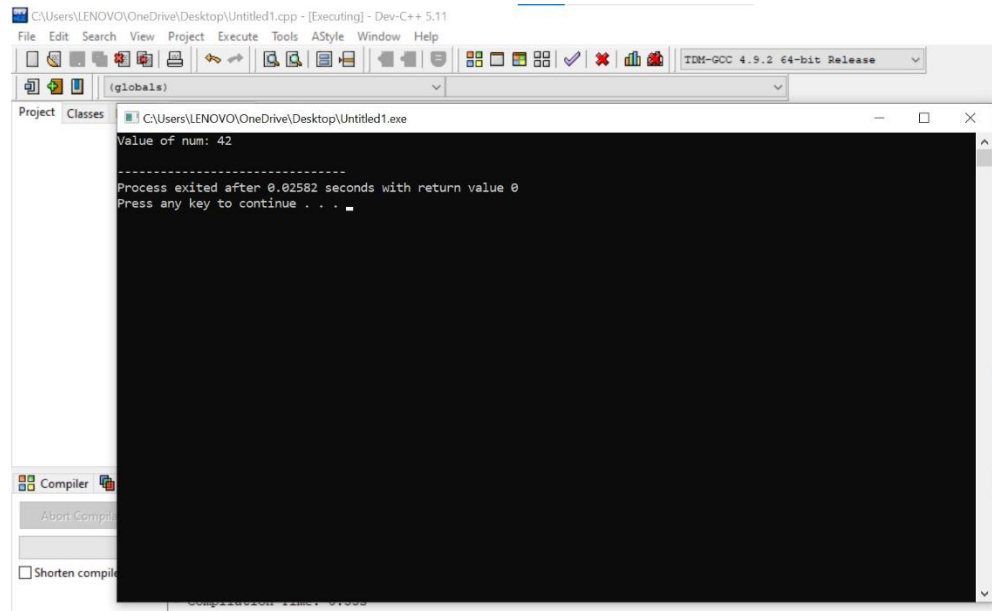
```
#include <iostream>
```

```
int main() { int num = 42; int *ptr1 = &num;
```

```
int **ptr2 = &ptr1; std::cout << "Value of num:
```

```
" << **ptr2 << std::endl;
```

```
}
```

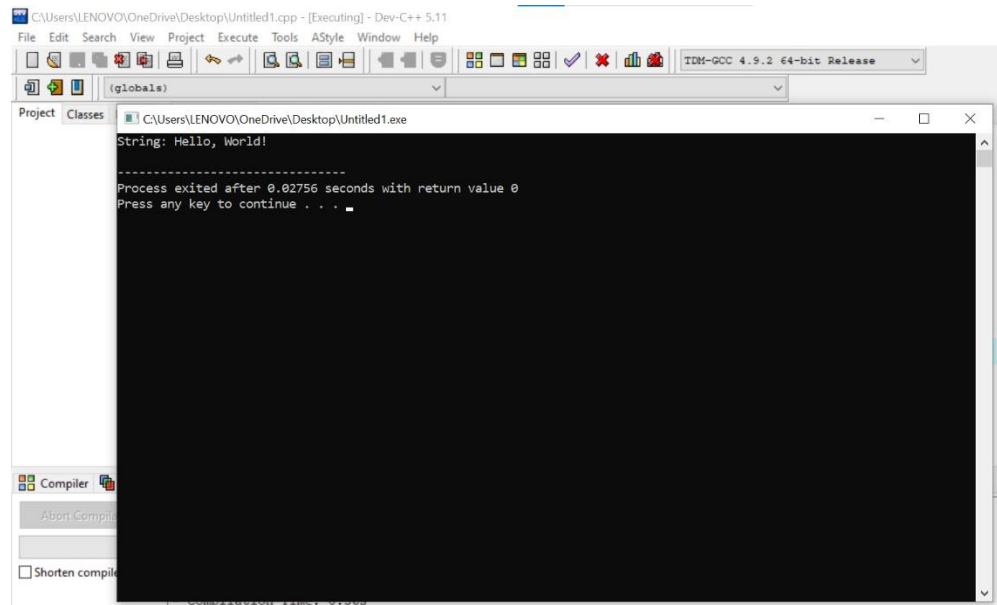


Program 10:

```
#include <iostream>
```

```
#include <cstring>
```

```
int main() {  const char *str = "Hello,  
World!";    std::cout << "String: " <<  
str << std::endl;  return 0;  
}
```



Program 11:

#include <iostream>

```
int main() {    const
```

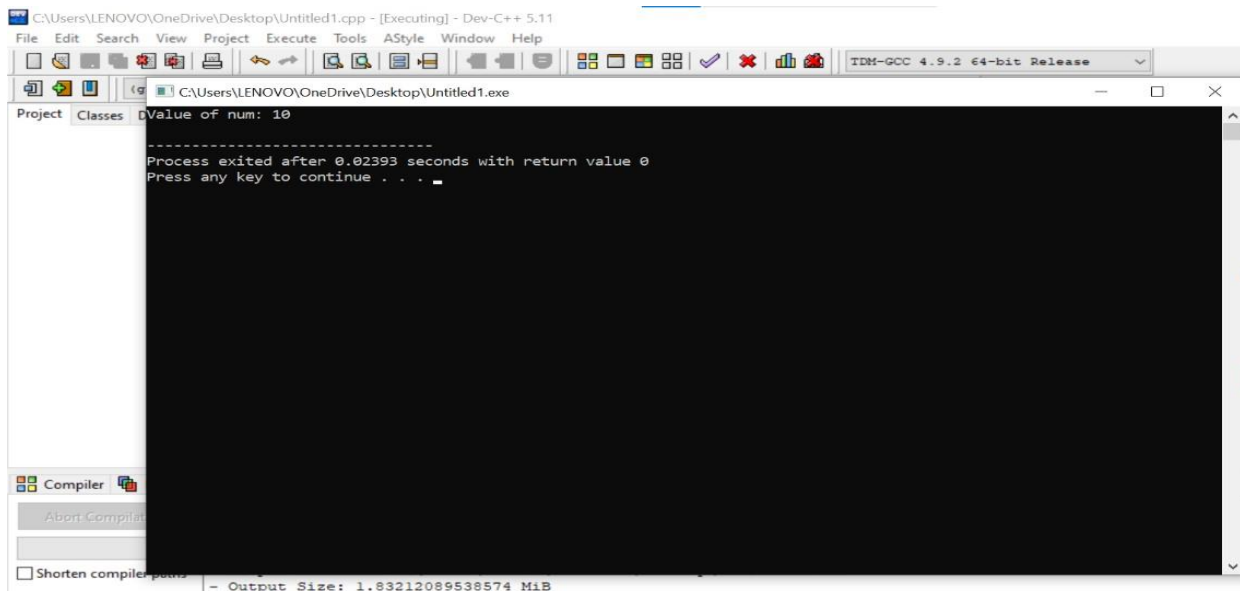
```
int num = 10;    const
```

```
int *ptr = &num;
```

```
    // *ptr = 20; // Error: Cannot modify constant data through ptr
```

```
    std::cout << "Value of num: " << *ptr << std::endl;
```

```
}
```



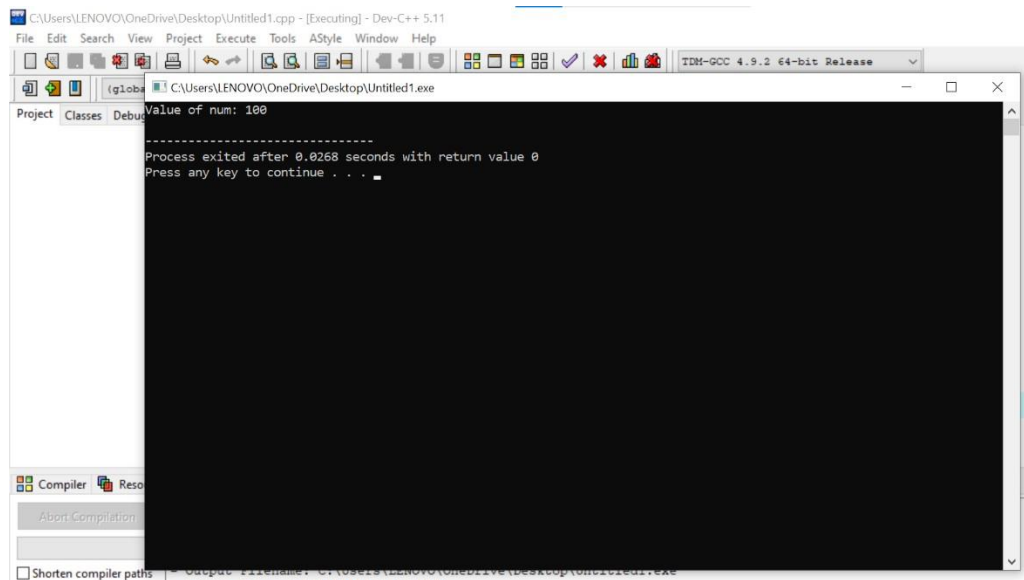
Program 12

```
#include <iostream>
```

```
int main() { int num  
= 42; int *const ptr  
= &num;
```

```
*ptr = 100; // OK

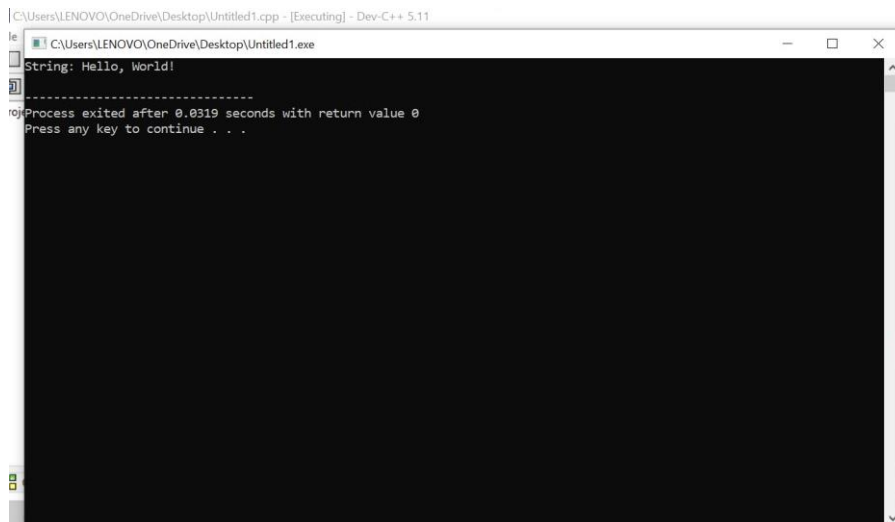
// ptr = nullptr; // Error: Cannot change the
pointer    std::cout << "Value of num: " << *ptr <<
std::endl;  return 0;
}
```



Program 13:

#include <iostream>

```
int main() { const char str[] = "Hello,  
World!";    const char *ptr = str;  
std::cout << "String: " << ptr <<  
std::endl;  return 0;  
}
```

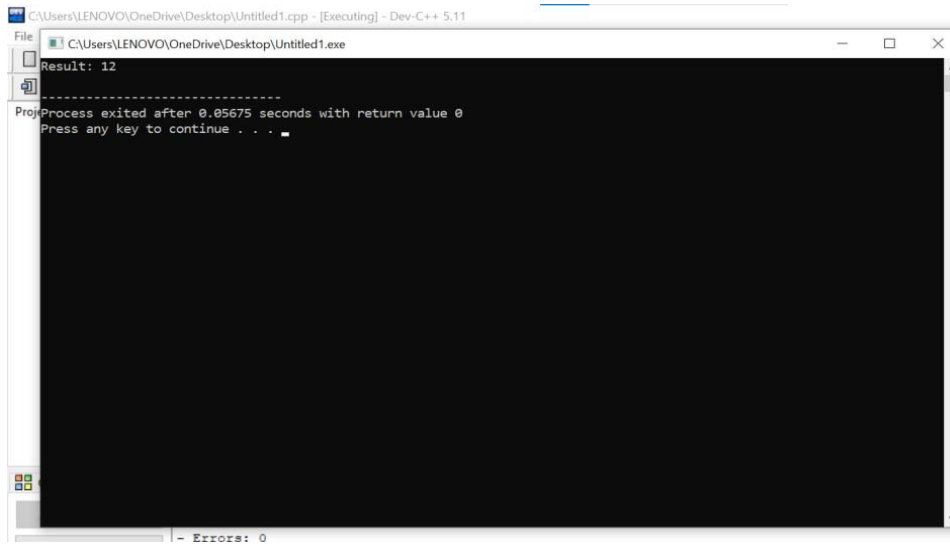


Program 14:

```
#include <iostream>
```

```
int add(int a, int b) {  
    return a + b;  
}
```

```
int main() {    int (*ptr)(int, int) = &add;  
    std::cout << "Result: " << (*ptr)(5, 7) <<  
    std::endl;    return 0;  
}
```



Program 15:

```
#include <iostream>
```

```
class MyClass {
```

```
public:
```

```
    int value;
```

```
    void setValue(int val) {
```

```
        value = val;
```

```
    }
```

```
};
```

```
int main() {
```

```
    MyClass    obj;
```

```
    MyClass *ptr = &obj;
```

```
    ptr->setValue(42);
```

```
std::cout << "Value:  
" << obj.value <<  
std::endl; return 0;  
}
```