

In the Name of Allah, the Most Gracious, the Most Merciful

COMSATS UNIVERSITY ISLAMABAD



Submitted By:

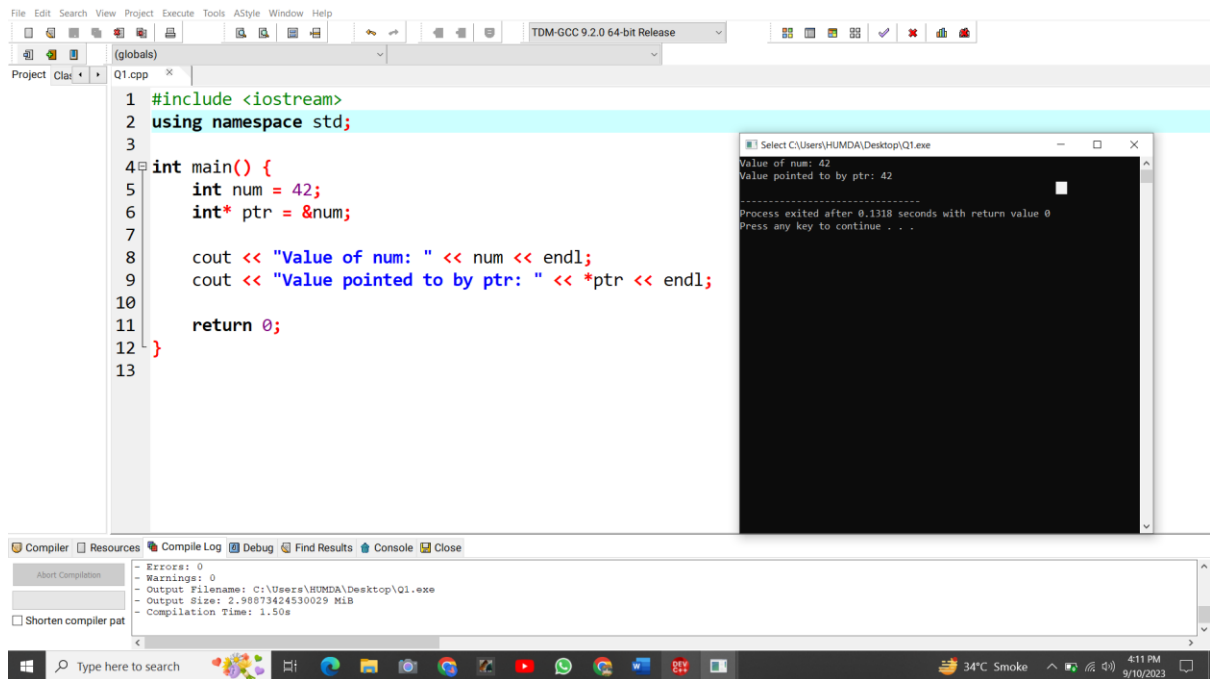
HUMDA TARIQ

Submitted To:

Sir Dewan Qaseem

- ❖ **Course:** Computer network(lab)
- ❖ **Assignment NO:** First
- ❖ **Registration No:** **SP22-BCS-128**

QUESTION NO:1



The screenshot shows a C++ IDE with a file named Q1.cpp. The code defines an integer variable 'num' with the value 42, and a pointer 'ptr' that points to 'num'. It then uses 'cout' to print the value of 'num' and the value pointed to by 'ptr'. The output window shows the results: 'Value of num: 42' and 'Value pointed to by ptr: 42'. The compiler output at the bottom indicates no errors or warnings.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int num = 42;
6     int* ptr = &num;
7
8     cout << "Value of num: " << num << endl;
9     cout << "Value pointed to by ptr: " << *ptr << endl;
10
11     return 0;
12 }
13
```

Compiler Output:

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\HUMDA\Desktop\Q1.exe
- Output Size: 2.98873424530029 MiB
- Compilation Time: 1.50s
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int num = 42;
```

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

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

```
    cout << "Value pointed to by ptr: " << *ptr << endl;
```

```
    return 0;
```

```
}
```

QUESTION NO:2

```
#include <iostream>
```

```
using namespace std;
```

```

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++) {

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

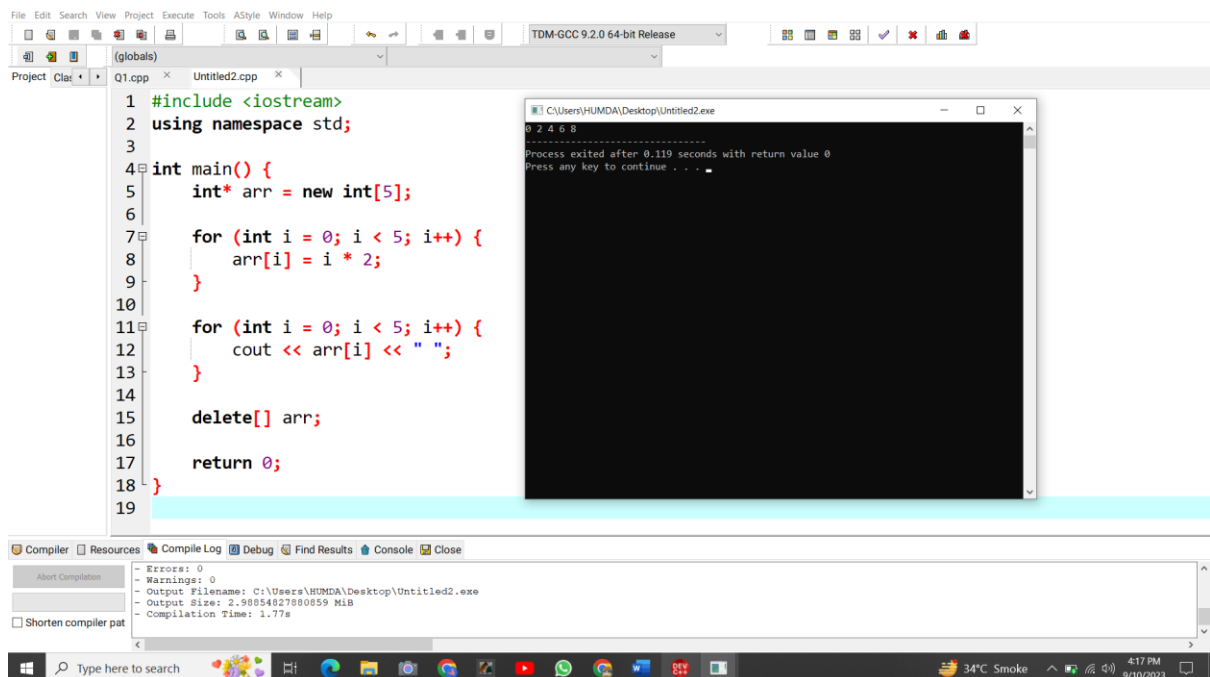
    }

    delete[] arr;

    return 0;

}

```



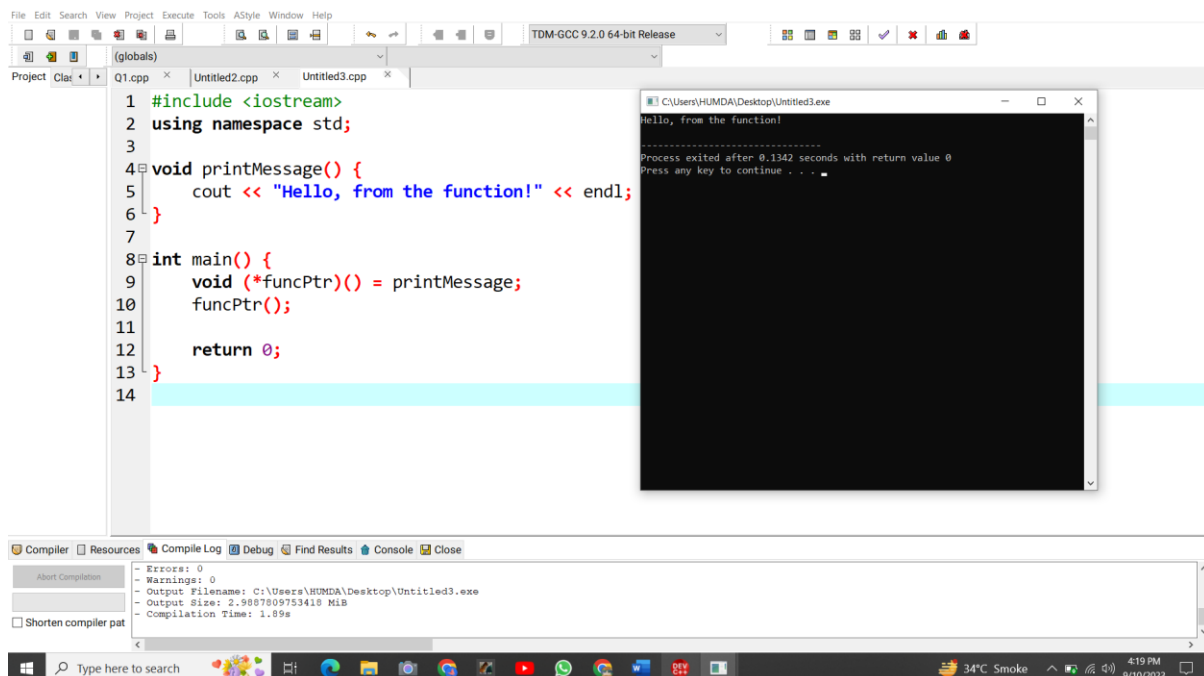
QUESTION NO:3

```
#include <iostream>
```

```
using namespace std;
```

```
void printMessage() {  
    cout << "Hello, from the function!" << endl;  
}
```

```
int main() {  
    void (*funcPtr)() = printMessage;  
    funcPtr();  
  
    return 0;  
}
```



QUESTION NO:4

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int num = 42;
```

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

```
    int** ptr2 = &ptr;
```

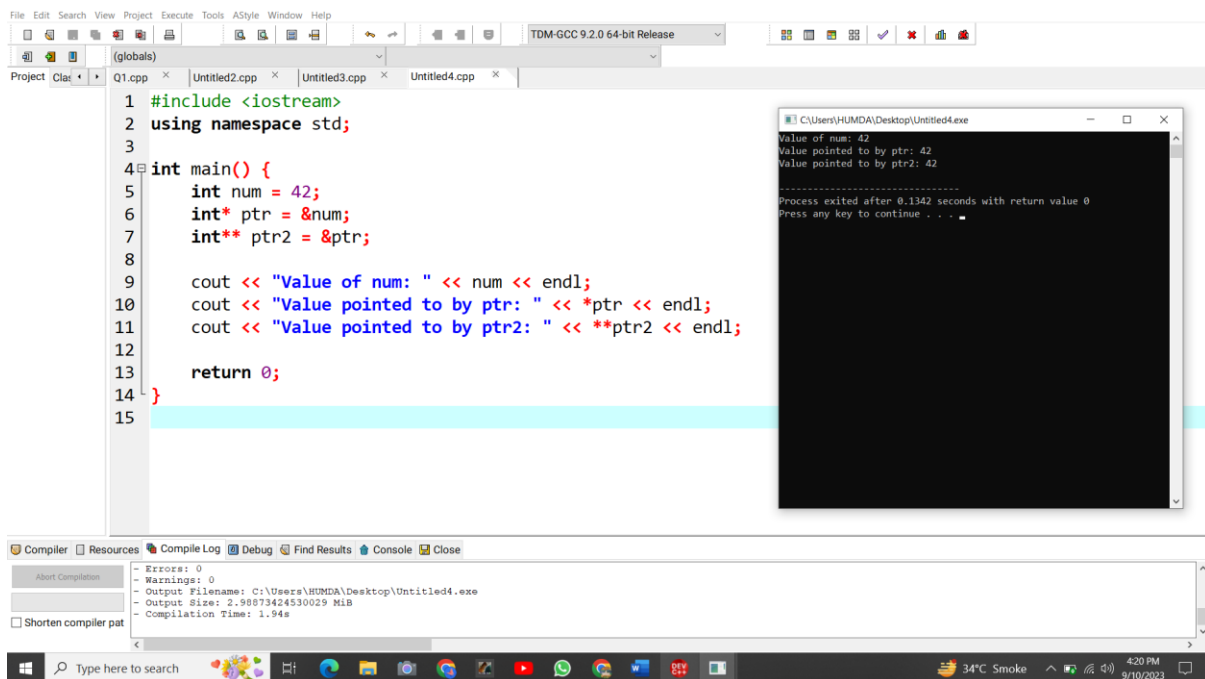
```
    cout << "Value of num: " << num << endl;
```

```
    cout << "Value pointed to by ptr: " << *ptr << endl;
```

```
    cout << "Value pointed to by ptr2: " << **ptr2 << endl;
```

```
    return 0;
```

```
}
```



QUESTION NO:5

```
#include <iostream>
```

```
using namespace std;
```

```
void modifyValue(int* ptr) {
```

```
    *ptr = 99;
```

```
}
```

```
int main() {
```

```
    int num = 42;
```

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

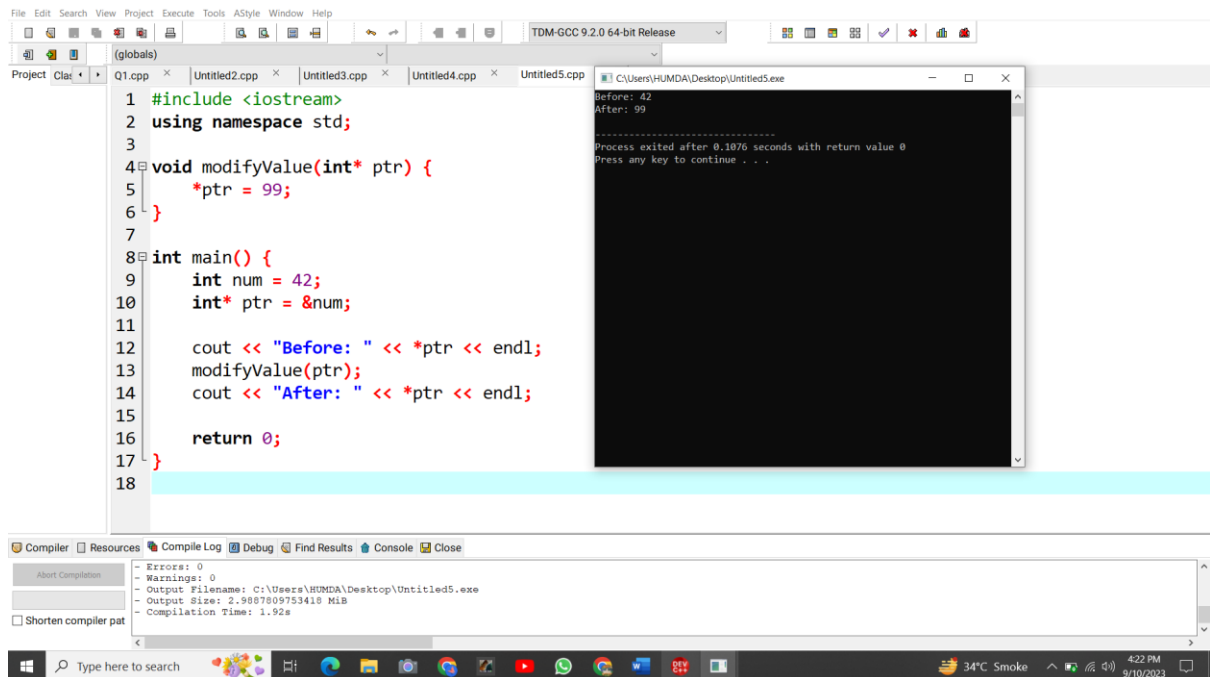
```
    cout << "Before: " << *ptr << endl;
```

```
    modifyValue(ptr);
```

```
    cout << "After: " << *ptr << endl;
```

```
    return 0;
```

```
}
```



QUESTION NO:6

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int arr[] = { 10, 20, 30, 40, 50};
```

```
    int* ptr = arr;
```

```
    for (int i = 0; i < 5; i++) {
```

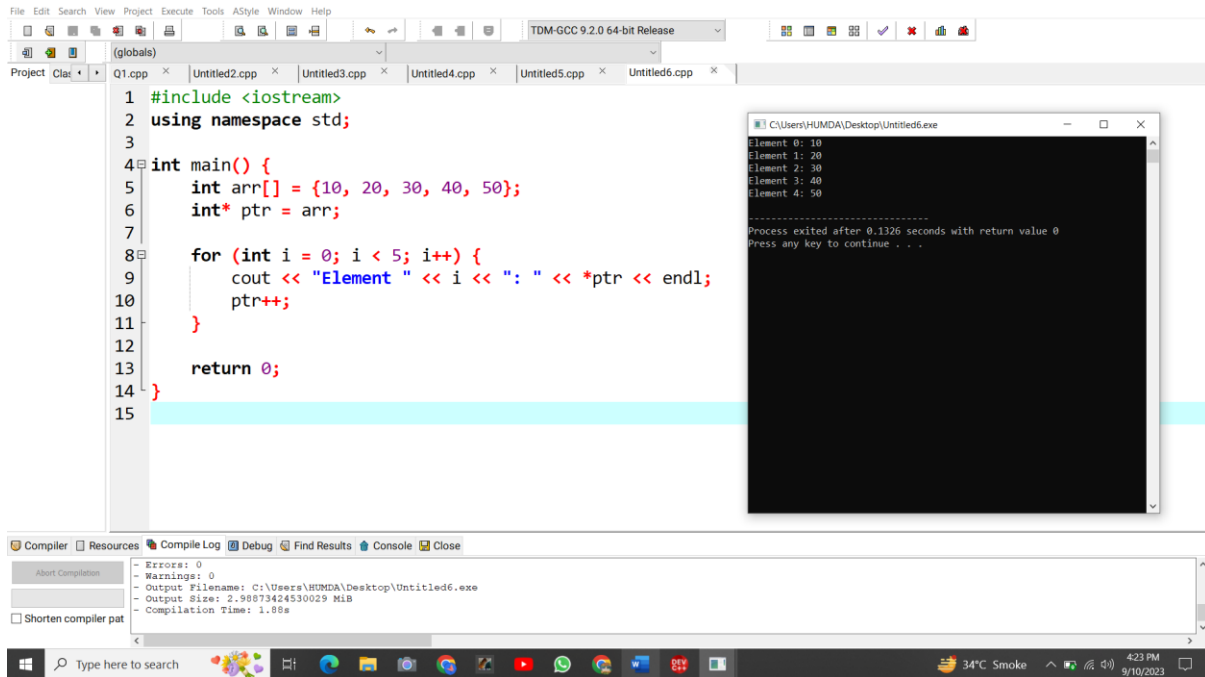
```
        cout << "Element " << i << ": " << *ptr << endl;
```

```
        ptr++;
```

```
    }
```

```
    return 0;
```

```
}
```



QUESTION NO:7

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int num1 = 42, num2 = 84, num3 = 126;
```

```
    int* arr[] = {&num1, &num2, &num3};
```

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

```
        cout << "Value " << i + 1 << ": " << *arr[i] << endl;
```

```
    }
```

```
    return 0;
```

```
}
```


The screenshot shows a C++ IDE with the following code in `Untitled7.cpp`:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int num1 = 42, num2 = 84, num3 = 126;
6     int* arr[] = {&num1, &num2, &num3};
7
8     for (int i = 0; i < 3; i++) {
9         cout << "Value " << i + 1 << ": " << *arr[i] << endl;
10    }
11
12    return 0;
13 }
```

The output window shows the following output:

```
Value 1: 42
Value 2: 84
Value 3: 126
-----
Process exited after 0.132 seconds with return value 0
Press any key to continue . . .
```

The compiler log shows the following output:

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\HUMDA\Desktop\Untitled7.exe
- Output Size: 2.98873424530029 MiB
- Compilation Time: 1.78s
```

QUESTION NO:8

The screenshot shows a C++ IDE with the following code in `Untitled8.cpp`:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     const int num = 42;
6     const int* ptr = &num;
7
8     cout << "Value of num: " << num << endl;
9     cout << "Value pointed to by ptr: " << *ptr << endl;
10
11    return 0;
12 }
13
```

The output window shows the following output:

```
Value of num: 42
Value pointed to by ptr: 42
-----
Process exited after 0.1291 seconds with return value 0
Press any key to continue . . .
```

The compiler log shows the following output:

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\HUMDA\Desktop\Untitled8.exe
- Output Size: 2.98873424530029 MiB
- Compilation Time: 1.89s
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    const int num = 42;
```

```
const int* ptr = &num;

cout << "Value of num: " << num << endl;
cout << "Value pointed to by ptr: " << *ptr << endl;

return 0;
}
```

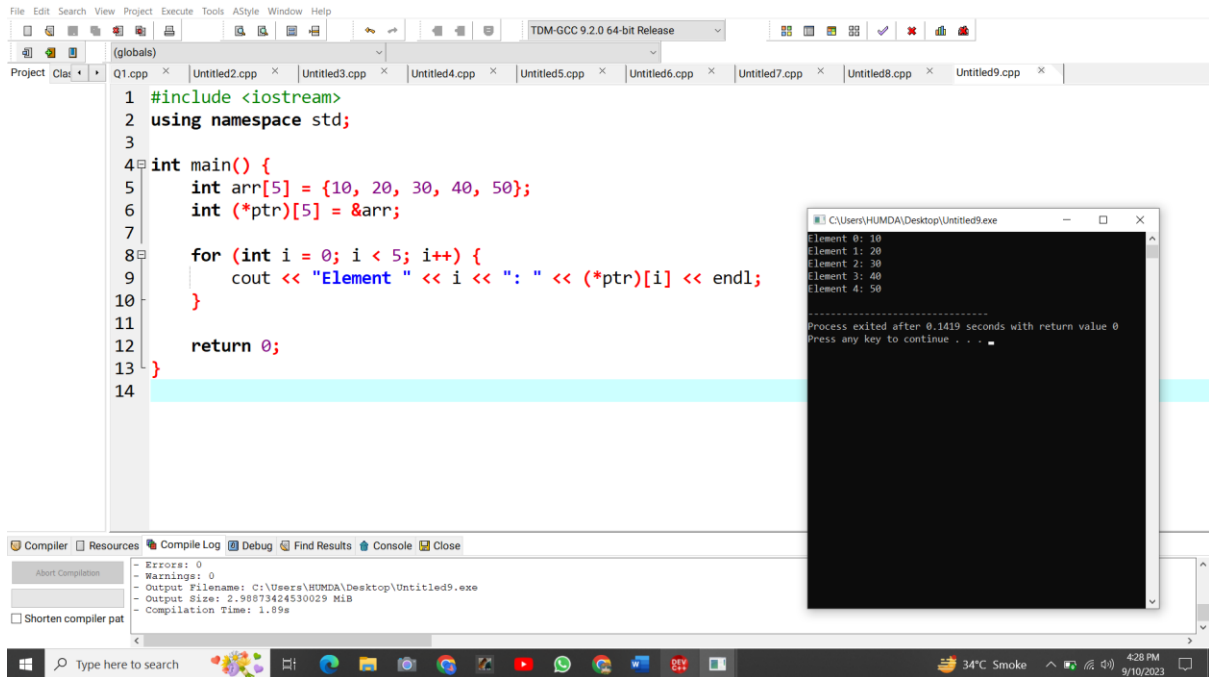
QUESTION NO:9

```
#include <iostream>
using namespace std;

int main() {
    int arr[5] = {10, 20, 30, 40, 50};
    int (*ptr)[5] = &arr;

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

    return 0;
}
```



QUESTION NO:10

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    char str[] = "Hello, World!";
```

```
    char* ptr = str;
```

```
    while (*ptr != '\0') {
```

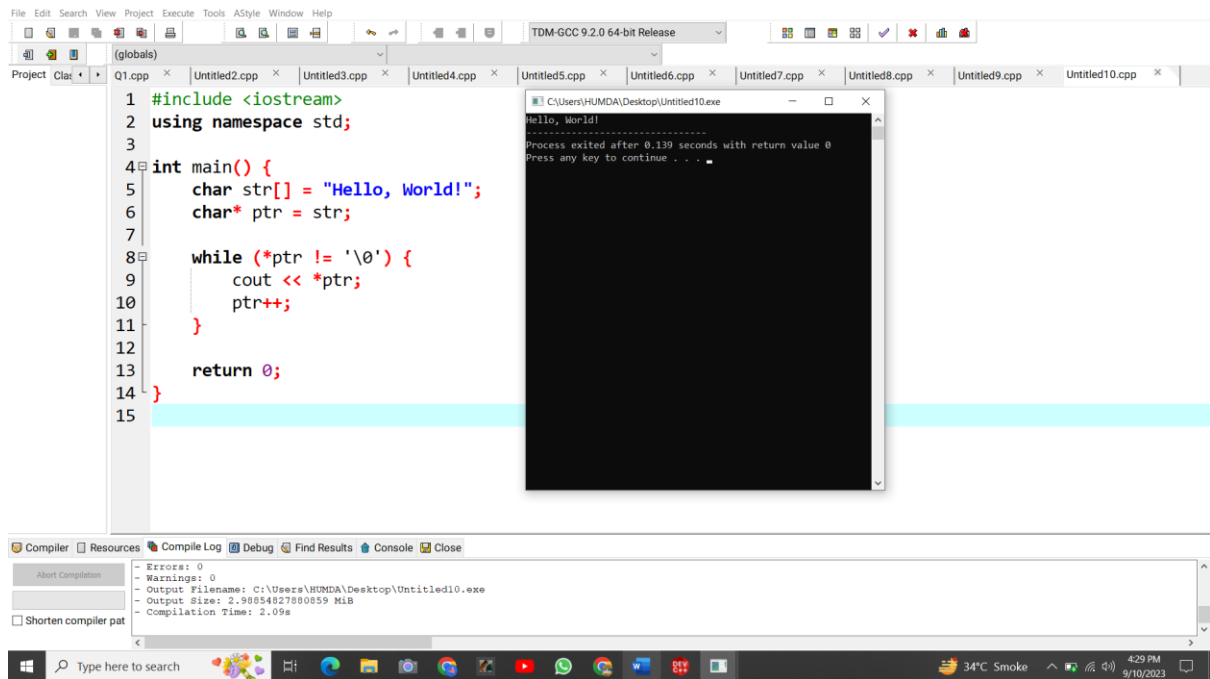
```
        cout << *ptr;
```

```
        ptr++;
```

```
    }
```

```
    return 0;
```

```
}
```



QUESTION NO:11

```
#include <iostream>
```

```
using namespace std;
```

```
class Rectangle {
```

```
public:
```

```
    int length;
```

```
    int width;
```

```
    Rectangle(int l, int w) : length(l), width(w) {}
```

```
};
```

```
int main() {
```

```
    Rectangle rect(5, 10);
```

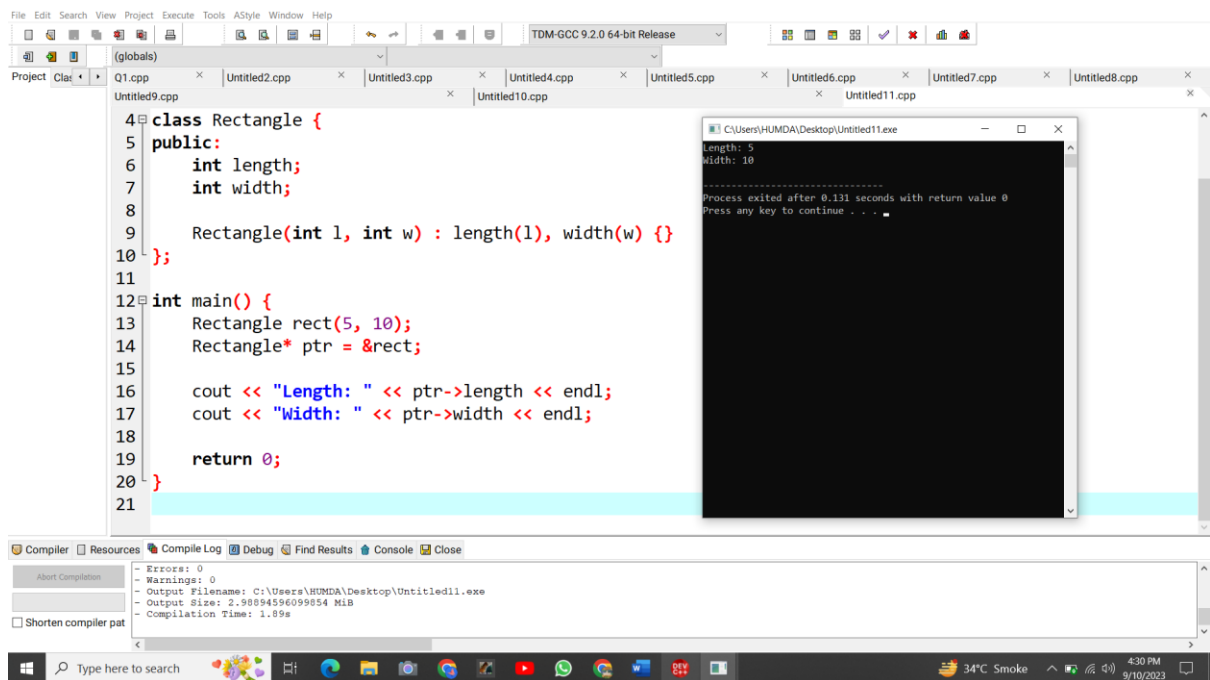
```
Rectangle* ptr = &rect;
```

```
cout << "Length: " << ptr->length << endl;
```

```
cout << "Width: " << ptr->width << endl;
```

```
return 0;
```

```
}
```



QUESTION NO:12

```
#include <iostream>
```

```
using namespace std;
```

```
class Calculator {
```

```
public:
```

```
int add(int a, int b) {
```

```
return a + b;
```

```

    }

    int subtract(int a, int b) {

        return a - b;

    }

};

int main() {

    Calculator calc;

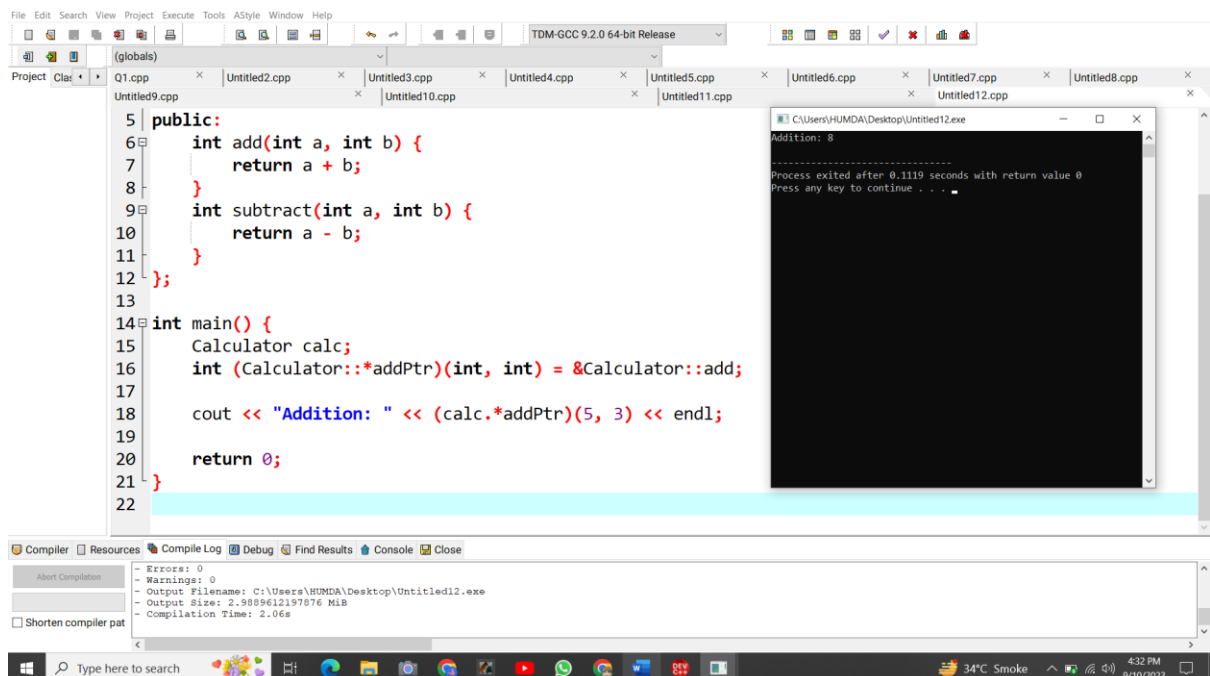
    int (Calculator::*addPtr)(int, int) = &Calculator::add;

    cout << "Addition: " << (calc.*addPtr)(5, 3) << endl;

    return 0;

}

```



QUESTION NO:13

```
#include <iostream>
```

```
using namespace std;
```

```
class Person {
```

```
public:
```

```
    string name;
```

```
    int age;
```

```
};
```

```
int main() {
```

```
    Person person;
```

```
    Person* ptr = &person;
```

```
    ptr->name = "Alice";
```

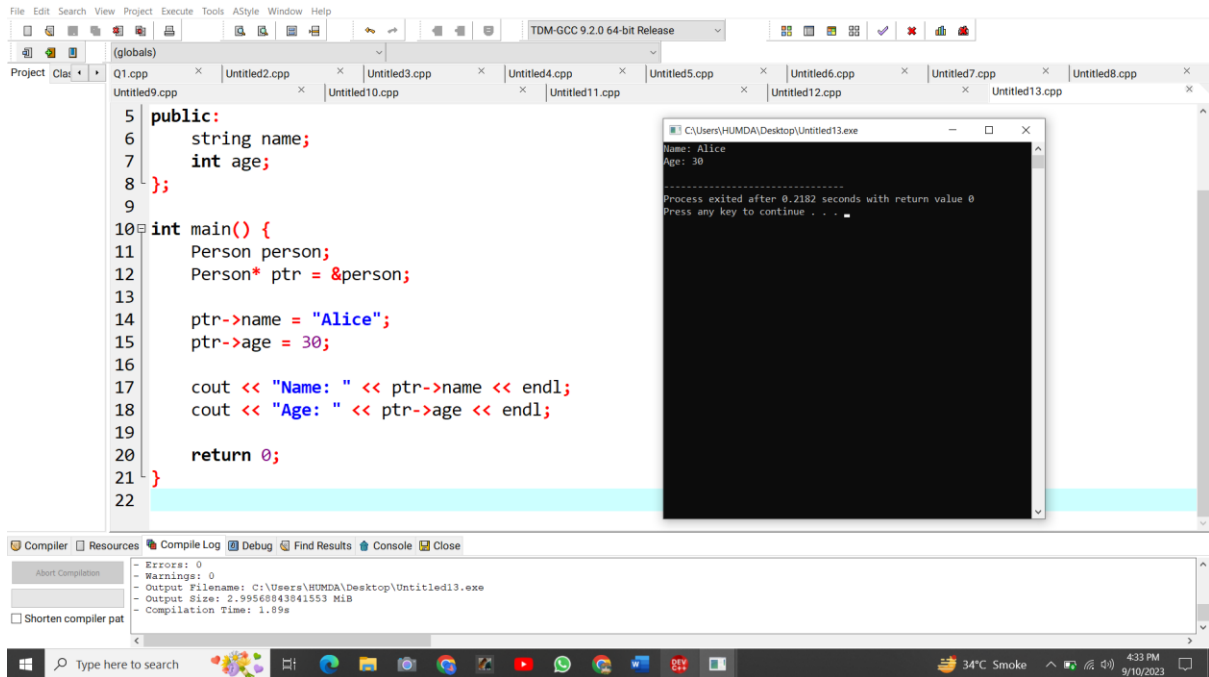
```
    ptr->age = 30;
```

```
    cout << "Name: " << ptr->name << endl;
```

```
    cout << "Age: " << ptr->age << endl;
```

```
    return 0;
```

```
}
```



QUESTION NO:14

```
#include <iostream>
```

```
using namespace std;
```

```
class Shape {
```

```
public:
```

```
    virtual void draw() {
```

```
        cout << "Drawing a shape." << endl;
```

```
    }
```

```
};
```

```
class Circle : public Shape {
```

```
public:
```

```
    void draw() override {
```

```
        cout << "Drawing a circle." << endl;
```



```

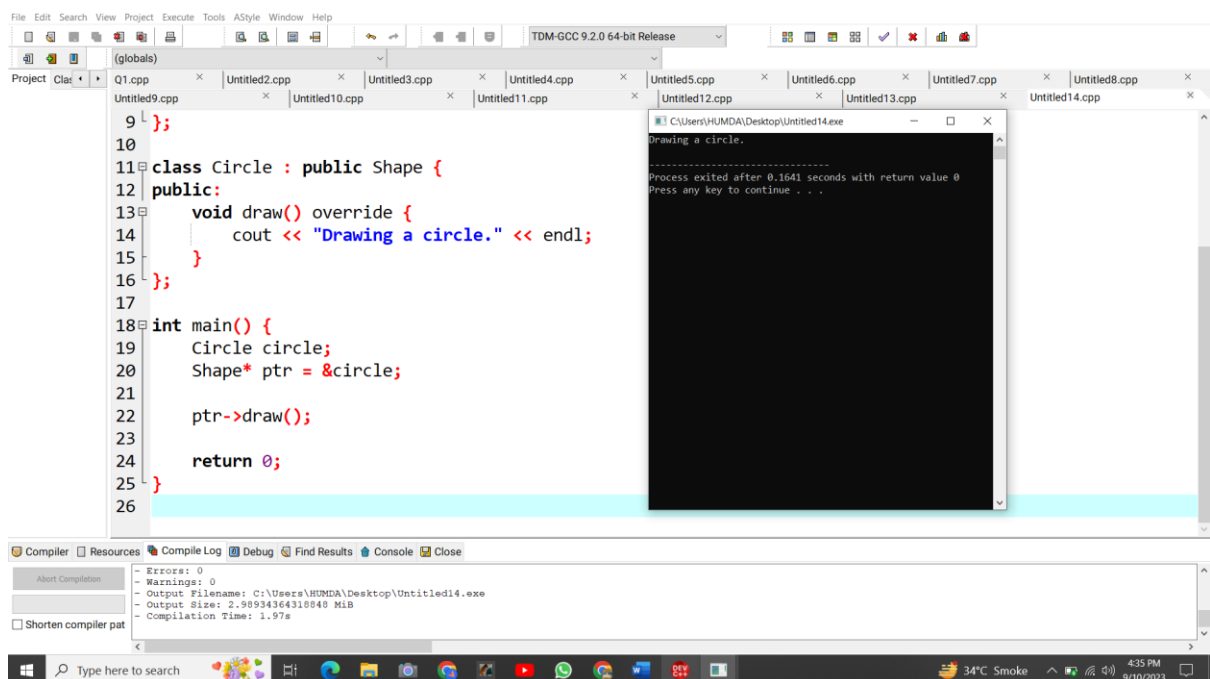
    }
};

int main() {
    Circle circle;
    Shape* ptr = &circle;

    ptr->draw();

    return 0;
}

```



QUESTION NO:15

```

#include <iostream>
using namespace std;

```

```
int main() {  
  
    int num = 42;  
  
    double pi = 3.14159;  
  
  
    void* ptr1 = &num;  
  
    void* ptr2 = &pi;  
  
  
    cout << "Integer value: " << *(int*)ptr1 << endl;  
    cout << "Double value: " << *(double*)ptr2 << endl;  
  
  
    return 0;  
  
}
```

