

## **CS 222**

### **Assignment 6 - Modular division**

**Name – Gautam Kumar Mahar**

**Roll No. – 2103114**

**Branch – Computer Science Engineering**

---

**extendedEuclid(int a, int b, int& x, int& y, int& gcd) -**

**Extended Euclid (int a, int b, int & x, int & y):** Using the extended Euclidean technique, this function determines the greatest common factor as well as the values of x and y in the equation  $ax + by = \text{gcd}(a, b)$ . The extended Euclidean method is repeated until b equals 0 and has an  $O(\log(\max(a,b)))$  time complexity. As a result, this function has an  $O(\log(\max(a,b)))$  time complexity.

**Pair<int, int> divide(int a, int b) const -**

**returns the residual after dividing a by b.** This function's time complexity is  $O(1)$  since integer division has an  $O(1)$  time complexity.

**findInverse() -**

The time complexity of the findInverse function is  $O(\log(N))$ , where  $N$  is a constant value defined in the class. This is because the extendedEuclid function is called.

**operator/(noModN y) -**

The time complexity of the operator/ function is  $O(\log(N))$ , where  $N$  is a constant value defined in the class. This is because it calls the findInverse function, which has a time complexity of  $O(\log(N))$  as explained above.

**getValue() -**

The time complexity of the getValue function is  $O(1)$ , since it performs a constant number of operations.

**Overall Time Complexity →**

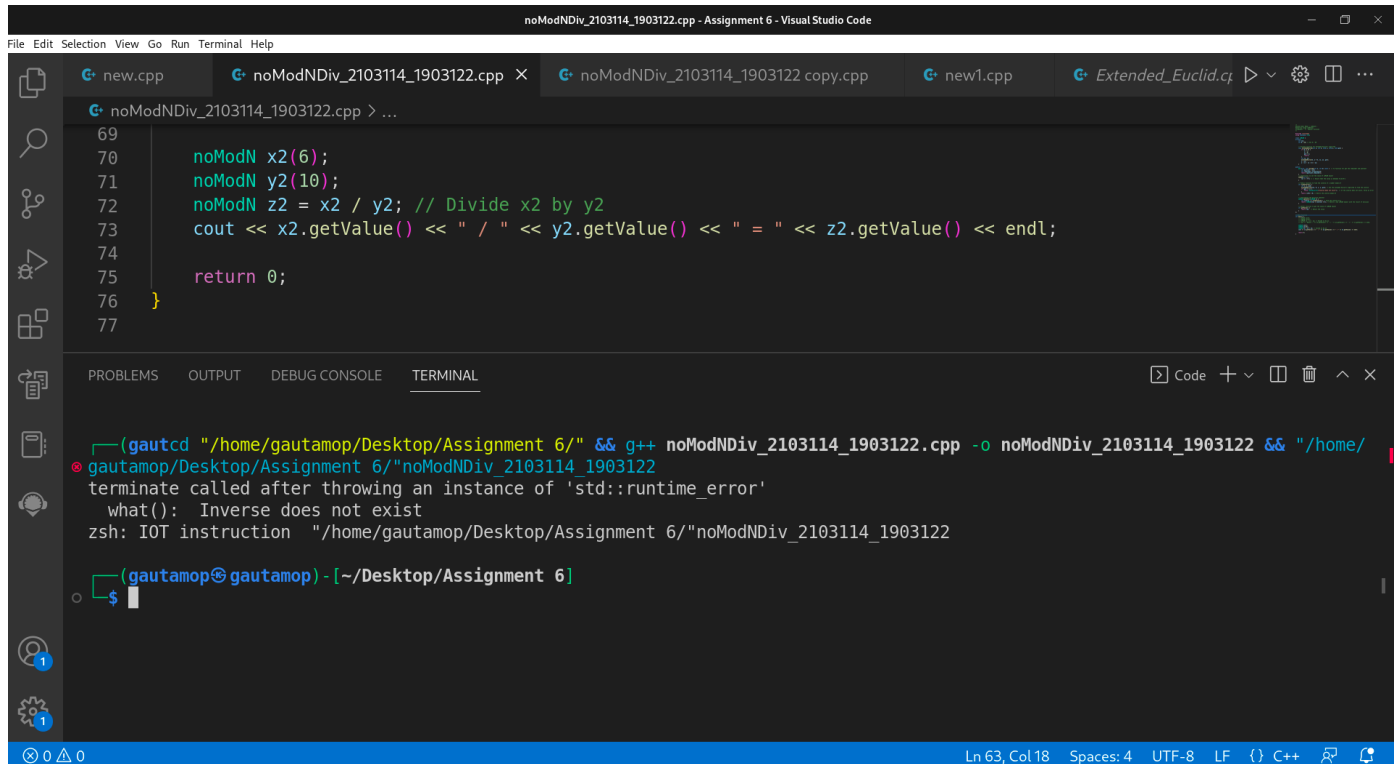
The number of times each function is called as a result determines the total temporal complexity of the code. The given main function only calls each function once, hence the code's overall time complexity is  $O(\log(\max(\text{value}, N)))$ .

**Output → (When N = 60) ← Given in this Question**

**$\gcd(10, 60) \neq 1$**

**Thats why it is showing error**

**because if  $\gcd \neq 1$  inverse modulo does not exist**



The screenshot shows the Visual Studio Code interface with a C++ file named `noModNDiv_2103114_1903122.cpp` open. The code defines a `noModN` struct with a `getValue()` method and a `main` function that calculates `z2 = x2 / y2` and prints the result. The terminal shows the command to compile and run the program, followed by a runtime error: `terminate called after throwing an instance of 'std::runtime_error'` with the message `what(): Inverse does not exist`. The error occurs because the modular inverse does not exist when the gcd is not 1.

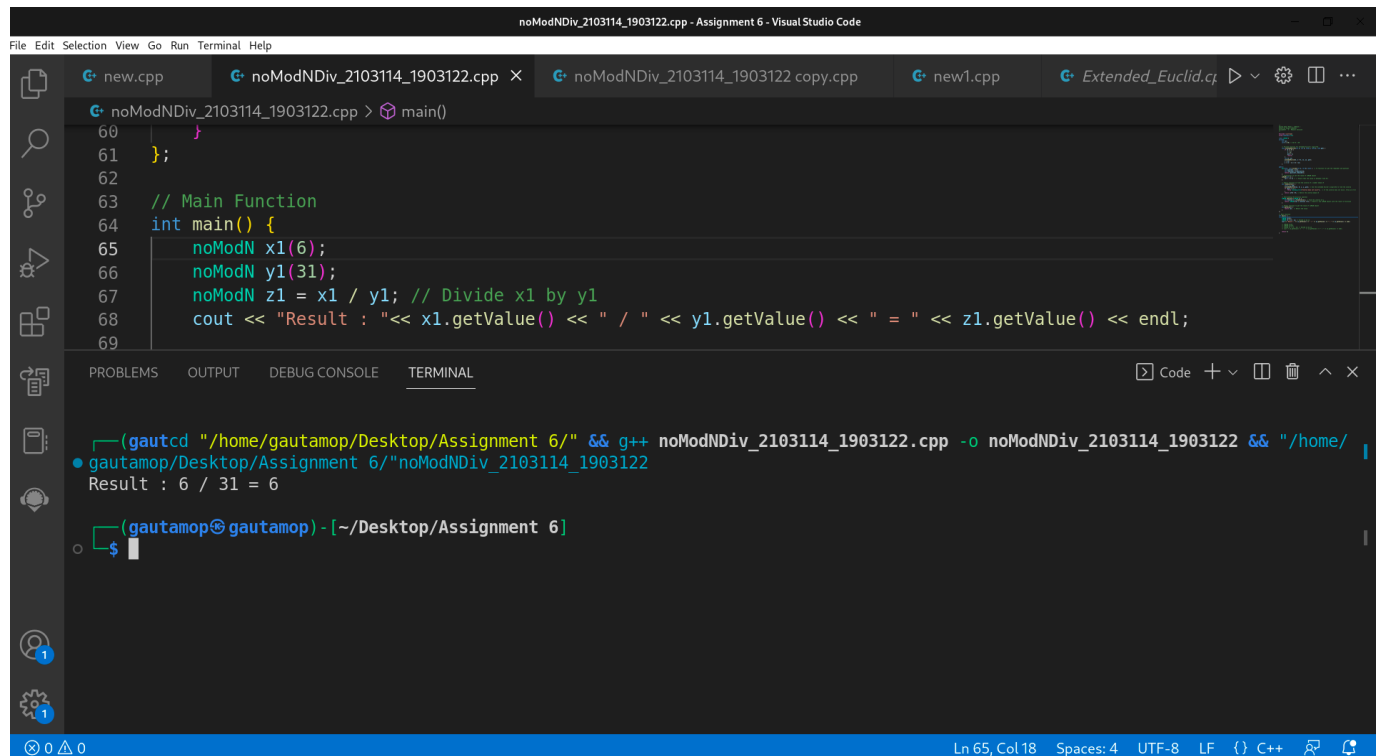
```
noModNDiv_2103114_1903122.cpp - Assignment 6 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

noModNDiv_2103114_1903122.cpp > ...
69
70     noModN x2(6);
71     noModN y2(10);
72     noModN z2 = x2 / y2; // Divide x2 by y2
73     cout << x2.getValue() << " / " << y2.getValue() << " = " << z2.getValue() << endl;
74
75     return 0;
76 }
77

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code + - - - - -
(gautcd "/home/gautamop/Desktop/Assignment 6/" && g++ noModNDiv_2103114_1903122.cpp -o noModNDiv_2103114_1903122 && "/home/
gautamop/Desktop/Assignment 6/"noModNDiv_2103114_1903122
terminate called after throwing an instance of 'std::runtime_error'
what(): Inverse does not exist
zsh: IOT instruction "/home/gautamop/Desktop/Assignment 6/"noModNDiv_2103114_1903122

(gautamop@gautamop) - [~/Desktop/Assignment 6]
$
```

it is equal to 31. So  $6 \times 31 = 186$  and  $186 \% 60 = 6$ .



## Thank You