CS 222
Assignment 6 - Modular division

Name – Gautam Kumar Mahar
Roll No. – 2103114
Name – Kanwarraj Singh
Roll No. – 1903122
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 axe + by = 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() -

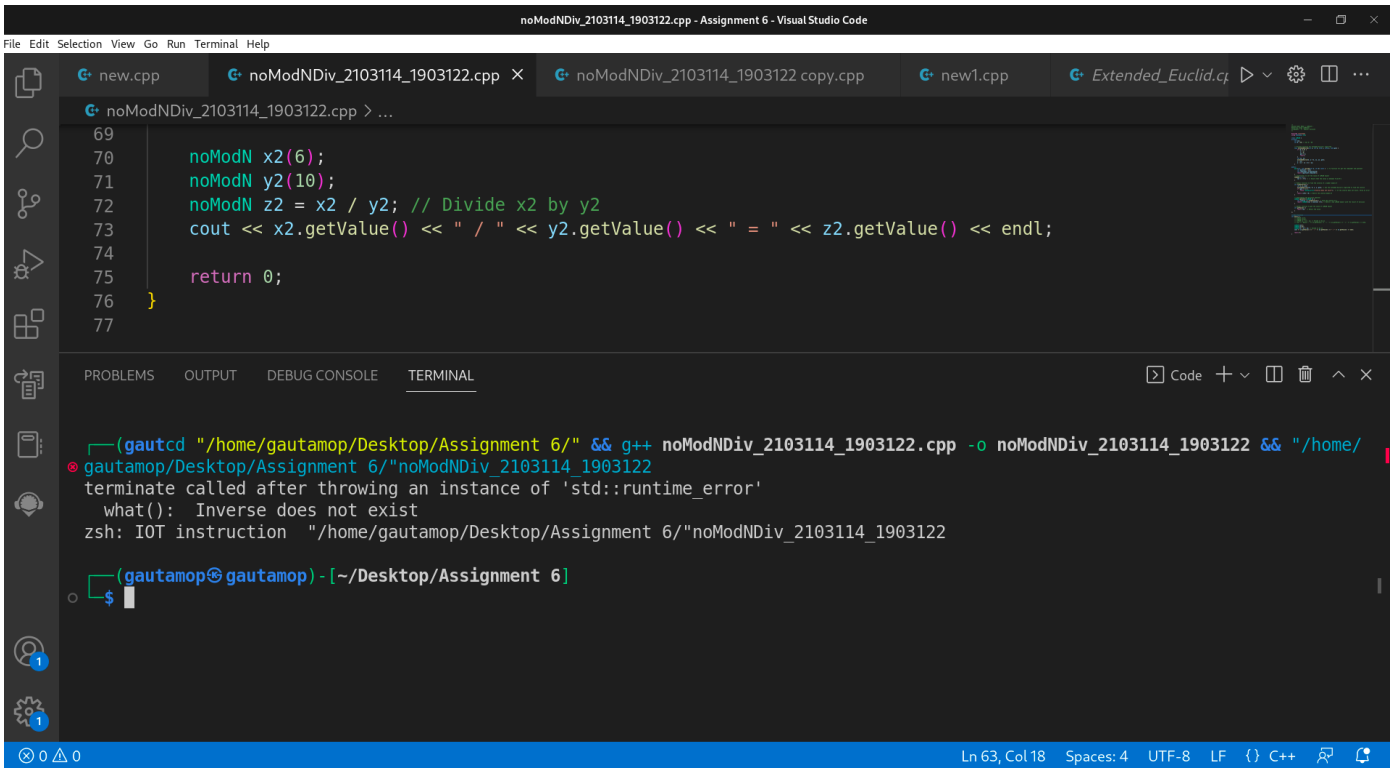
The time complexity of the getValue function is O(1), since it performs a constant number of operations.

Overall Time Complexity →

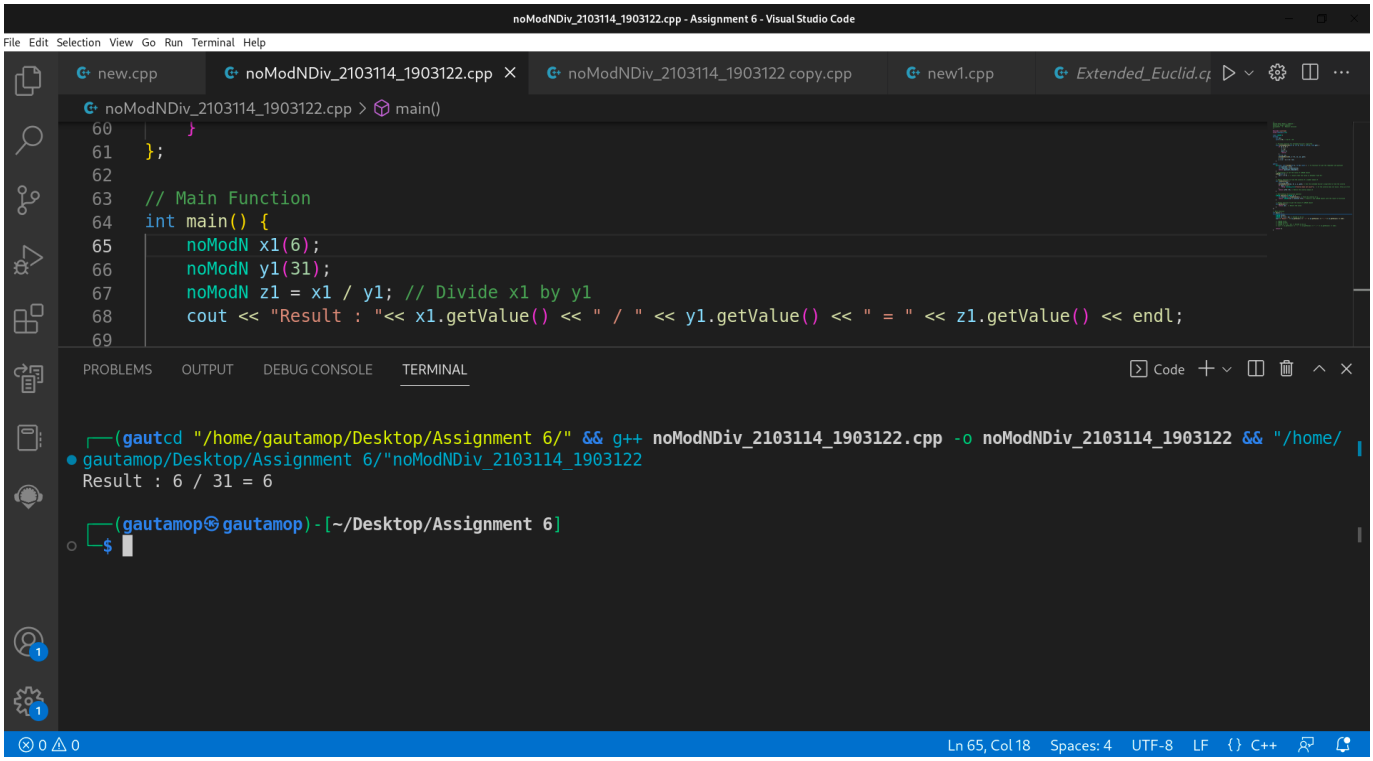
The number of times each function is called as a result determines the total temporal complexity of the code. The given main function only calls each function once, hence the code's overall time complexity is O(log(max(value, N)).

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

gcd(10,60) != 1
Thats why it is showing error
because if gcd!=1 inverse modulo does not exist



gcd(31,60)=1
Thats why inverse modulo exist and
it is equal to 31.So 6*31=186 and 186%60=6.



Thank You