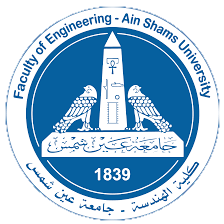
****

Course Name: Design and Analysis of Algorithms

Course Code: CSE332s

**Algorithms Project**

|  |  |
| --- | --- |
| **Name** | **ID** |
| **Manar Ahmed Mohamed omran** | **1806133** |

Team Member

**Task 6**

Assumptions

1. This algorithm works on exactly *12* coins.
2. The coins contain at most *1* fake coin.
3. The algorithm will always find a solution whether its identifying the fake or assuring they’re all genuine.

Problem Description

We have 12 coins that looks identical, yet either they are indeed identical or one is fake. The fake one is different in weight but we don’t know if its lighter or heavier. We can only use balance scale without weights to identify the fake coin in the minimum number of weighings as well as identifying if its lighter or heavier or identify they are all genuine. Using Dynamic Programing algorithm.

Pseudo Code & Solution Steps

Step1- split the coins into 3 groups of 4

Step2- weigh two groups against each other and store the balancing information determining

if the groups are balanced(all genuine) or one is heavier/lighter than the other

Step3- use the stored information of the subproblems to determine information of other coins

until identifying the fake coin

Complexity Analysis

The maximum number of weighings required to identify the fake coin is log3(n).

The time complexity of each weighing operation is O(n).

Therefore, the overall time complexity would be n\*log3(n) which is O(nlogn), where n is the number

of coins.

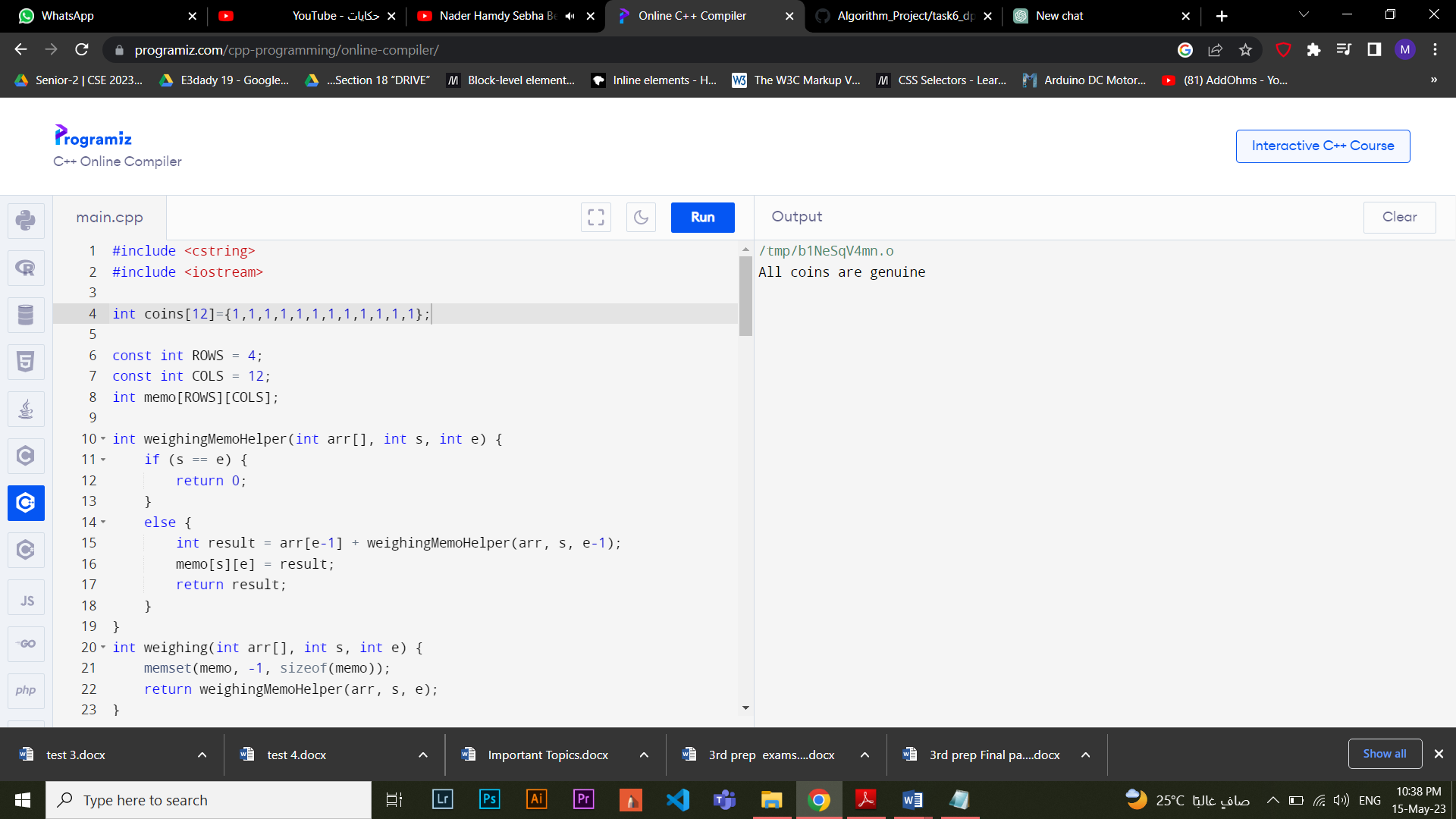
Comparison with another technique

Compared to brute force method, this algorithm archives less time complexity - as in brute force the time complexity equals to O(N), as it iterates over each coin in the worst case scenario. While by dynamic programming it does it in the minimum number of weighings with time complexity O(nlogn).

Output

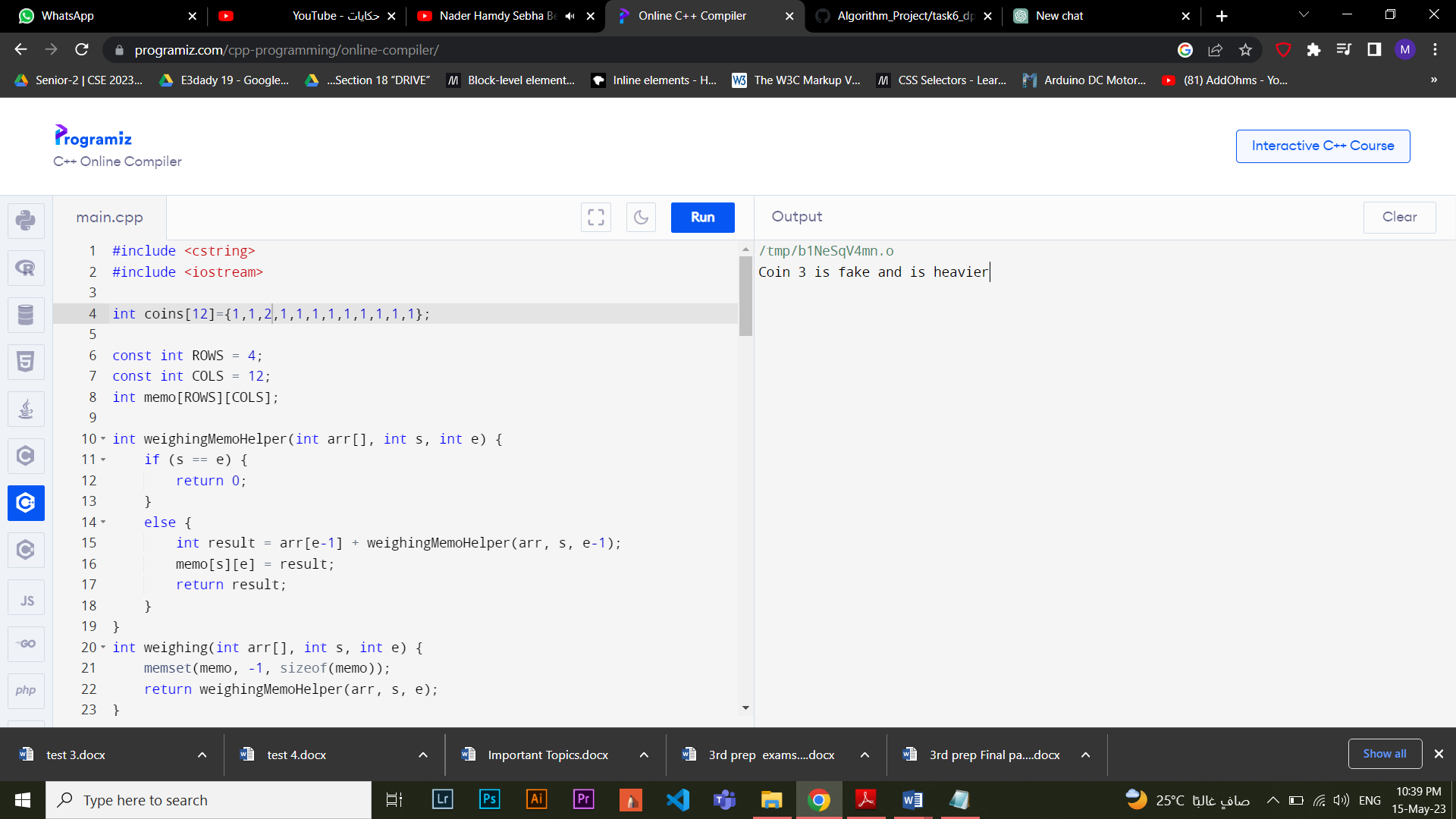
Example 1: coins[12]={1,1,1,1,1,1,1,1,1,1,1,1}

Expected output: All coins are Genuine.



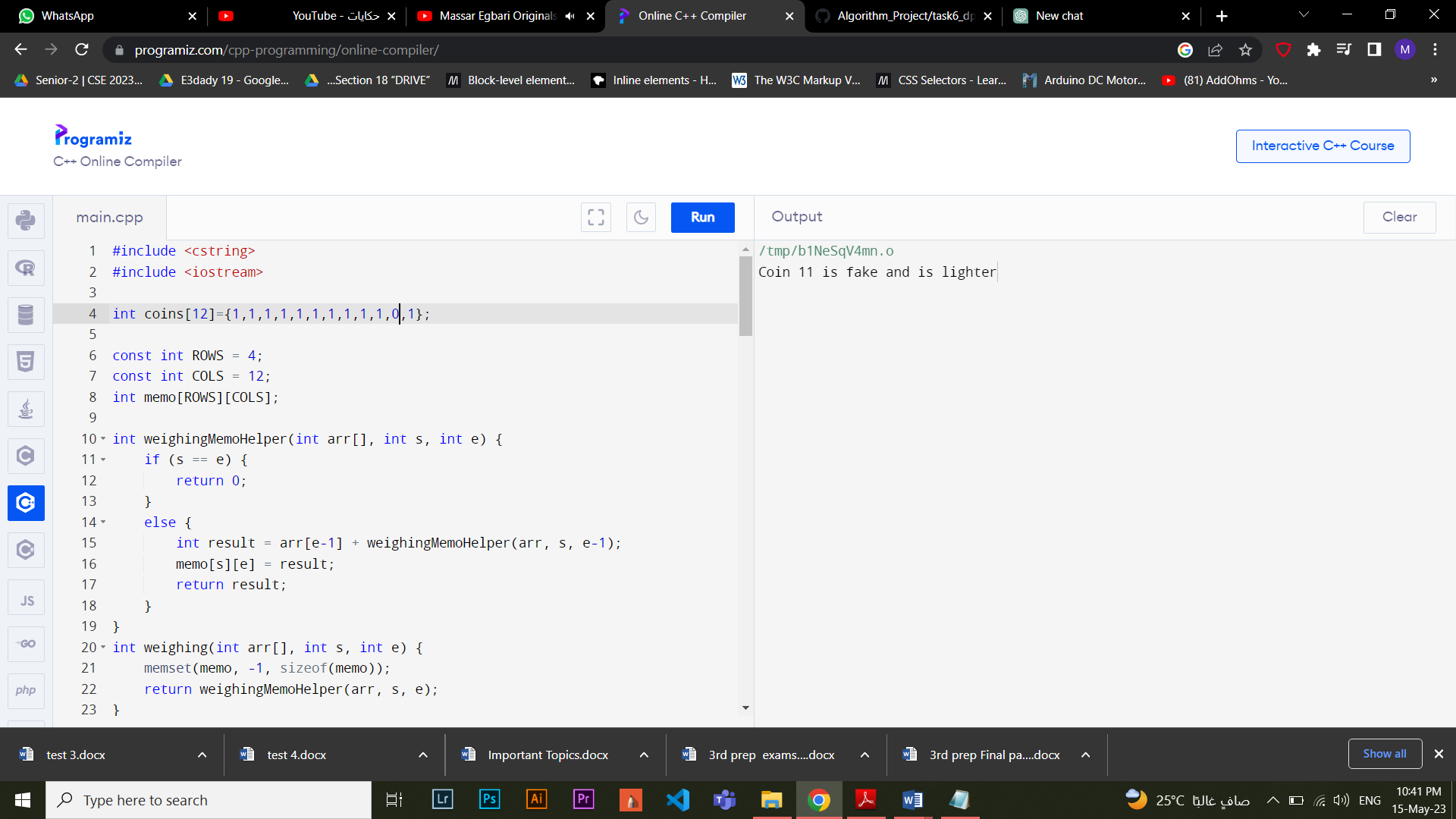
Example 2: coins[12]={1,1,2,1,1,1,1,1,1,1,1,1}

Expected output: Coin 3 is fake and is heavier



Example 3: coins[12]={1,1,1,1,1,1,1,1,1,1,0,1}

Expected output: Coin 11 is fake and is lighter.



Conclusion

Dynamic programming can solve the problem in the minimum number of weighings which is 3.

**References**  
1]

[**https://www.gregegan.net/SCIENCE/FindTheFakeCoin/FindTheFakeCoin.html**](https://www.gregegan.net/SCIENCE/FindTheFakeCoin/FindTheFakeCoin.html)

2]

[**https://www.geeksforgeeks.org/decision-trees-fake-coin-puzzle/**](https://www.geeksforgeeks.org/decision-trees-fake-coin-puzzle/)

3]

[**https://iq.opengenus.org/fake-coin-problem/#:~:text=Here%20is%20an%20algorithm%3A&text=If%20more%20than%20one%20coins,number%20of%20coins%20in%20C**](https://iq.opengenus.org/fake-coin-problem/#:~:text=Here%20is%20an%20algorithm%3A&text=If%20more%20than%20one%20coins,number%20of%20coins%20in%20C)**.**