SECURIN

PROBLEM STATEMENT: THE DOOMED DICE CHALLENGE

PART-A

1. How many total combinations are possible? Show the math along with the code !

Solution to the problem statement:

* By mathematical formula it is easy to find the total combinations

Total\_Combinations = len(dice\_A) \* len(dice\_B)

The total combinations are calculated by multiplying the number of faces on each die.

In this case, both dice have 6 faces each. Total Combinations = 6 (faces on Die A) \* 6 (faces on Die B) = 36 combinations.

1. Calculate and display the distribution of all possible combinations that can be obtained when rolling both Die A and Die B together. Show the math along with the code! Hint: A 6 x 6 Matrix.

To calculate the distribution of all possible combinations, we can create a 6x6 matrix where the rows represent the faces of Die A and the columns represent the faces of Die B. Then, iterate through all combinations and count the occurrences of each sum.

Fig 1.a – Input code

Fig 1.b – output ( It’ll print the output of both the problems­)

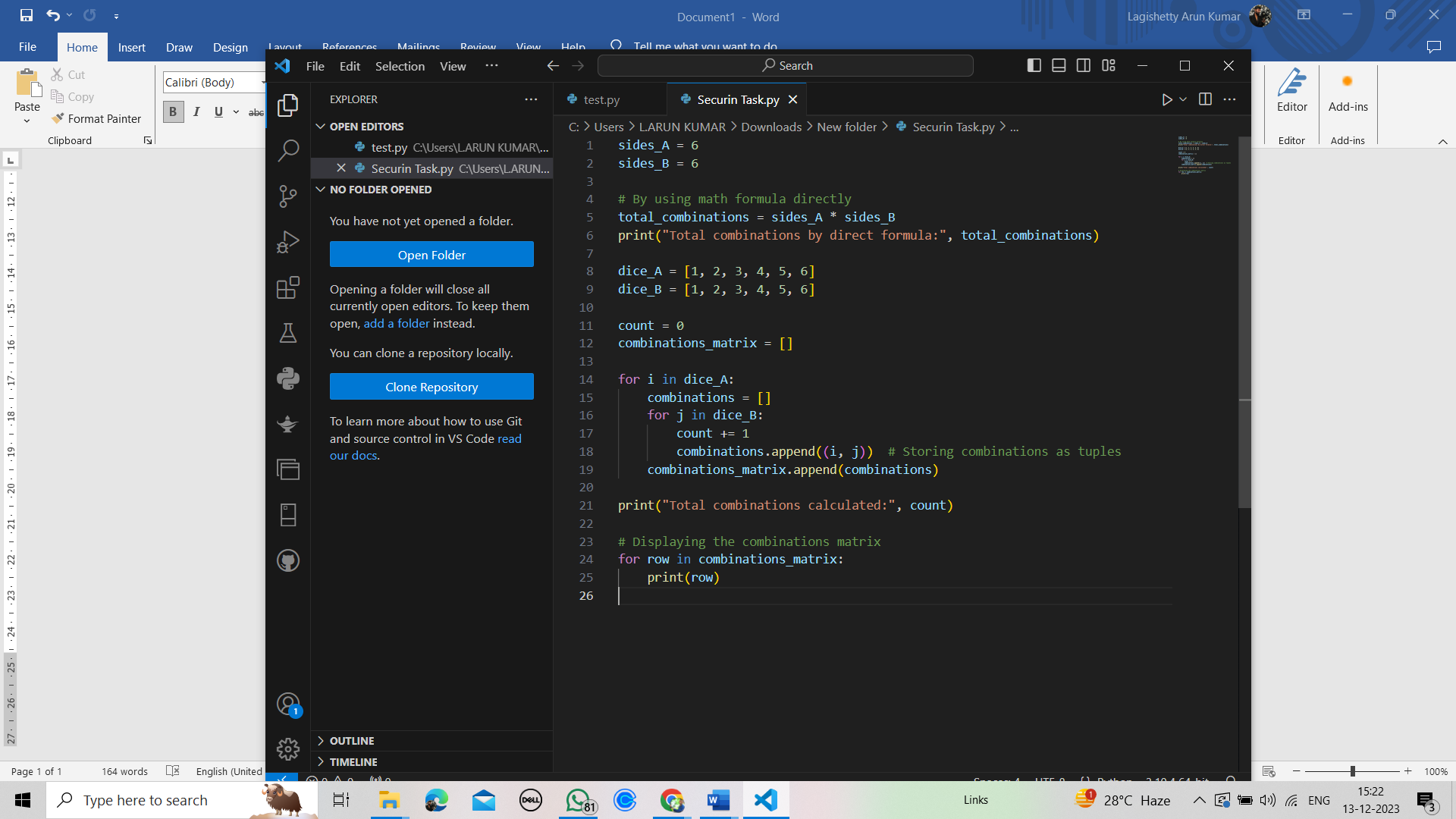


Fig 1.a

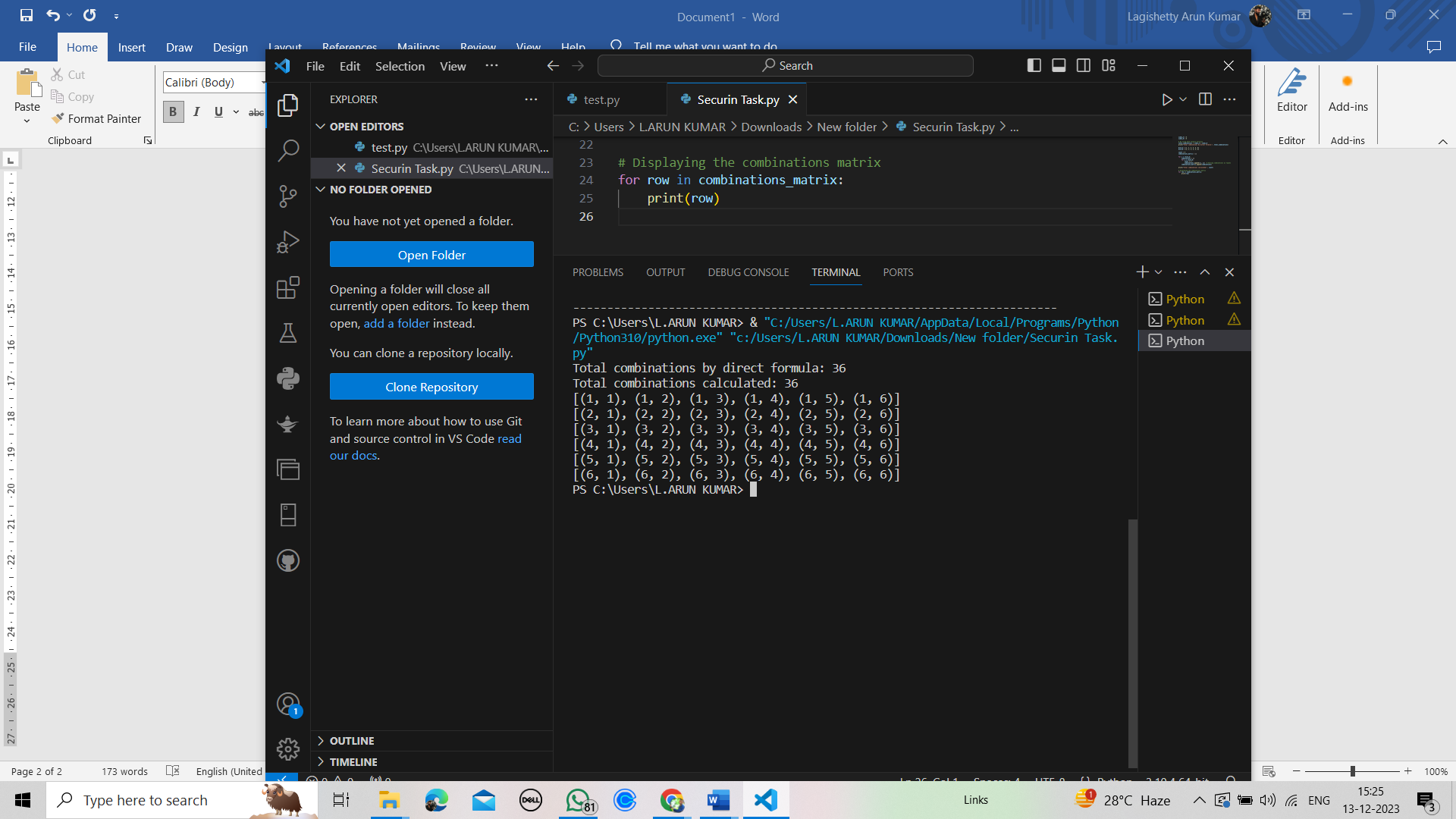


Fig 1.b

3 . Calculate the Probability of all Possible Sums occurring among the number of combinations from (2).

Explanation :

To the get the sum of combinations, I add the numbers of each combination and count the occurene of each sum.

To get the probability of each sum, I follow the below math equation :

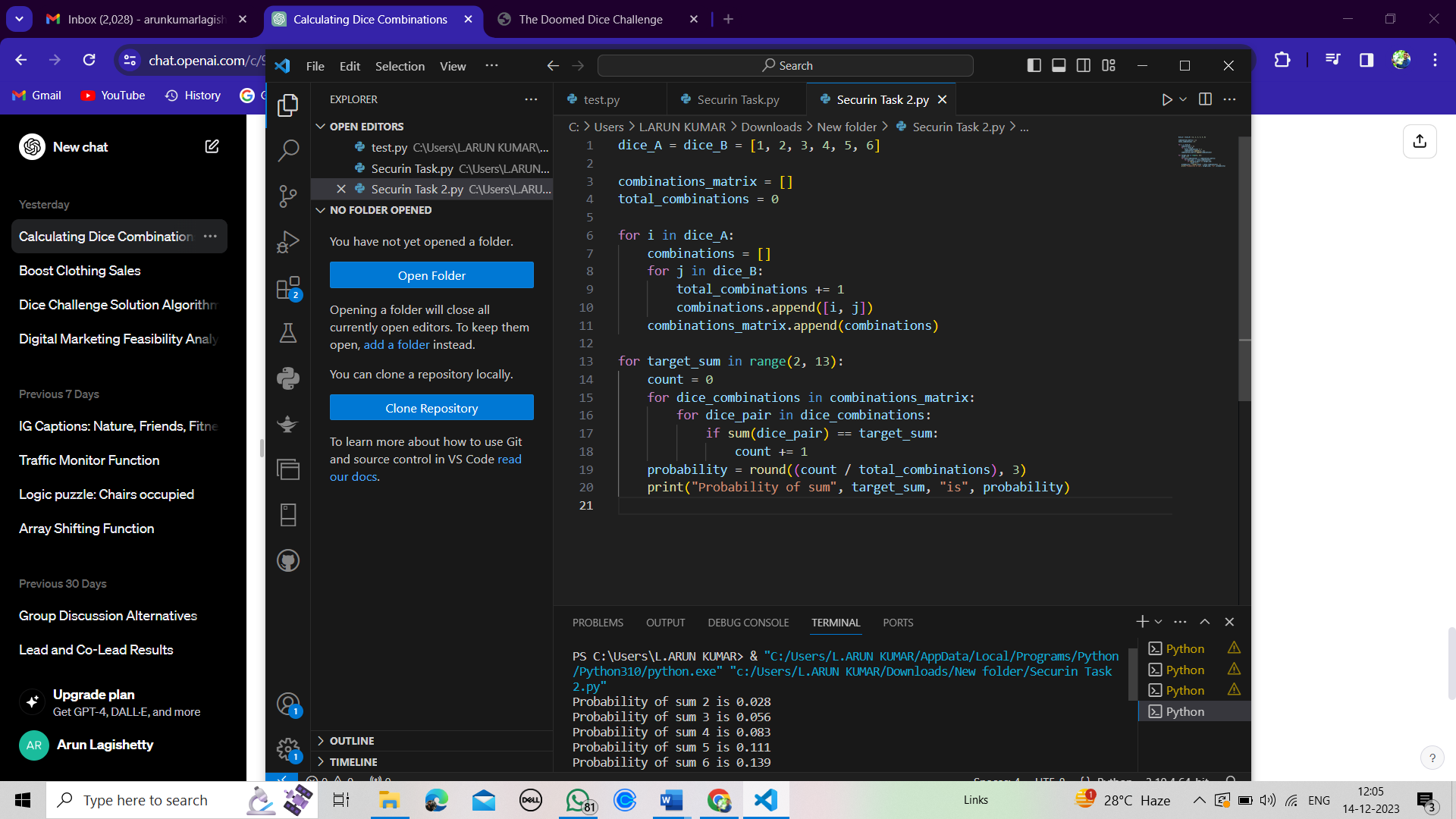
Probability = count of sum occurence / Total no.of combinations

* Firstly, I get the combinations\_matrix which contains all the combinations.
* After that we have to write another loop which is traversed through the range of between 2 and 13.
* For each range of between 2 and 13, We have to traverse through the every combination in the combination\_matrix.
* If the sum of numbers in the combination is equal to the range of current loop. We will increase the count by 1.
* After every loop we do the math formula to find the probability of the sum of occurences and print it

Probability = count / Total\_Combinations.

Fig 2-a --- input code

Fig 2-b --- output



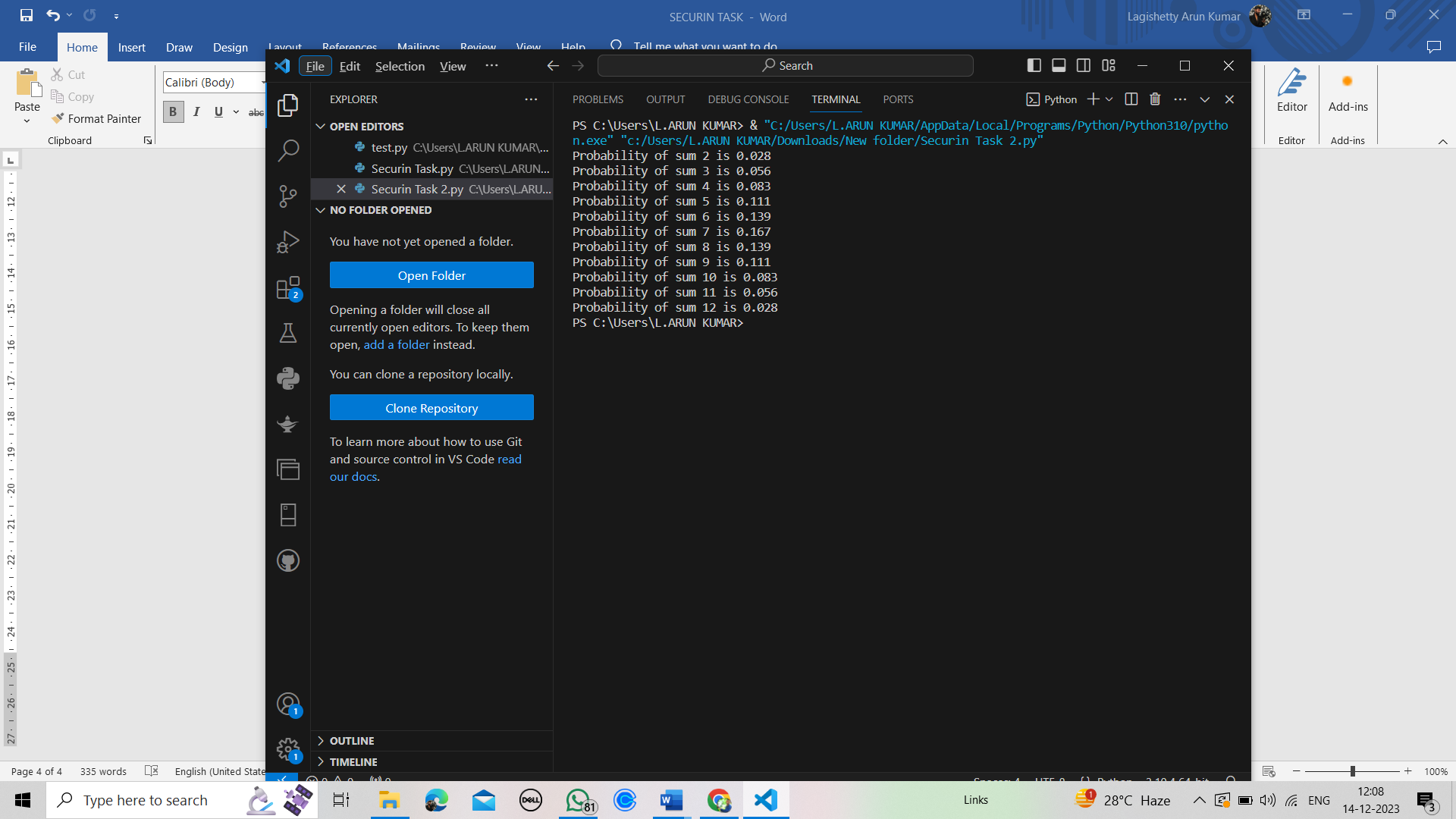
 Fig 2-a

Fig 2-b

Part-B

EXPLANATION FOR THE PROBLEM STATEMENT:

Firstly, the function undoom\_dice takes two input parameters : die\_A and die\_B, representing the faces of two dice. Then new\_die\_A and new\_die\_B are initialized as empty lists to store the new dice configurations that meet the conditions.

Now, count is set to zero and will be used to keep track of the number of valid combinations.

Then, Nested loops iterate through all combinations of faces from die\_A and die\_B. Within the loops, it checks if the value from die\_A is less than or equal to 4 ( it is the condition specified).

Additionally, it checks whether the sum of a value from die\_A and a value from die\_B is less than or equal to 6 (another specified condition).

If both conditions are met, it adds the corresponding values to new\_die\_A and new\_die\_B respectively, and increments the count. The function returns the newly generated dice configurations new\_die\_A and new\_die\_B. The function is called with initial dice configurations Die\_A and Die\_B.

The resulting new dice configurations New\_Die\_A and New\_Die\_B are printed to display the transformed dice faces that satisfy the specified conditions.

Fig 3-a --- Input code

Fig3-b --- Output code

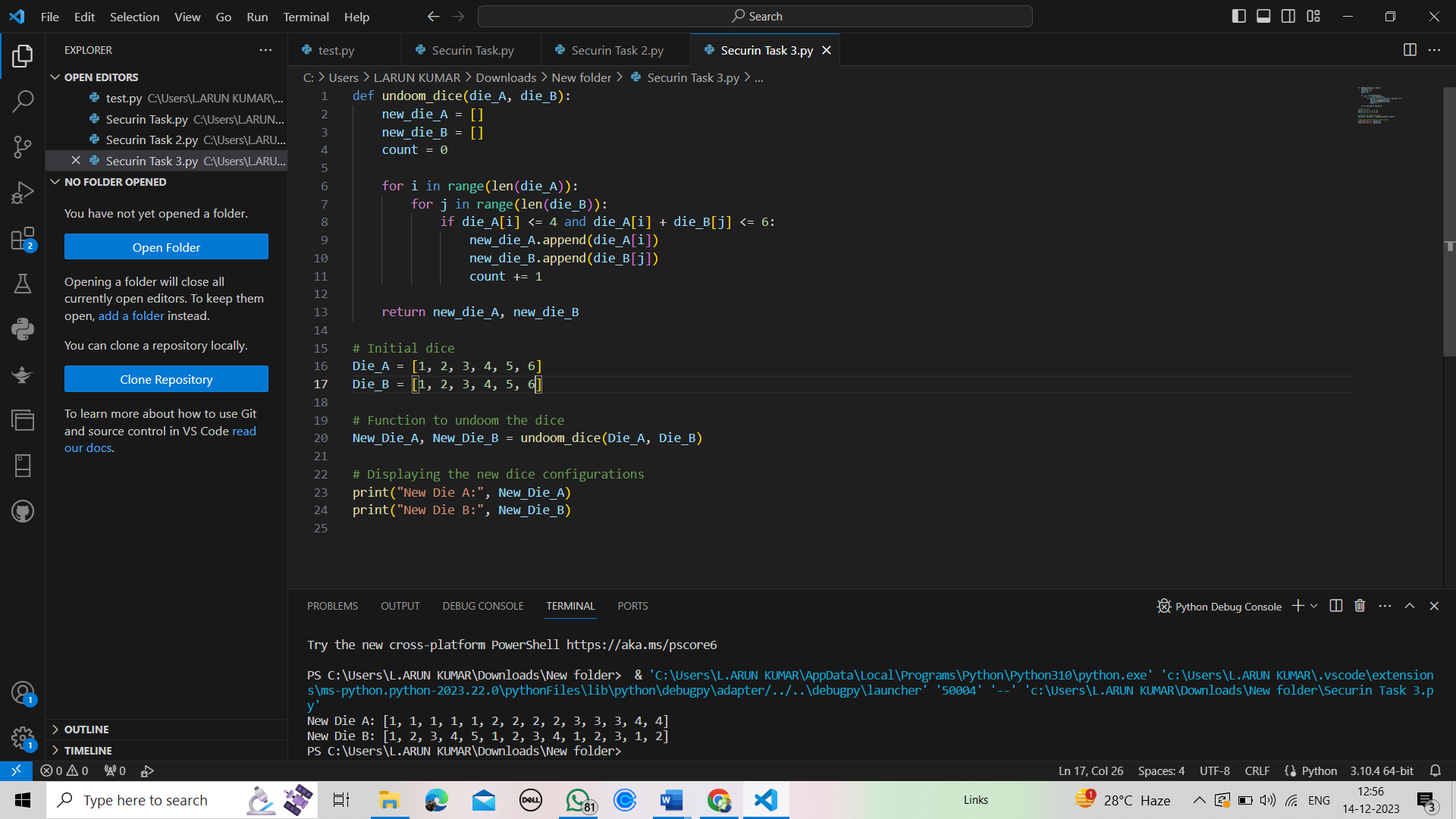


Fig. 3-a

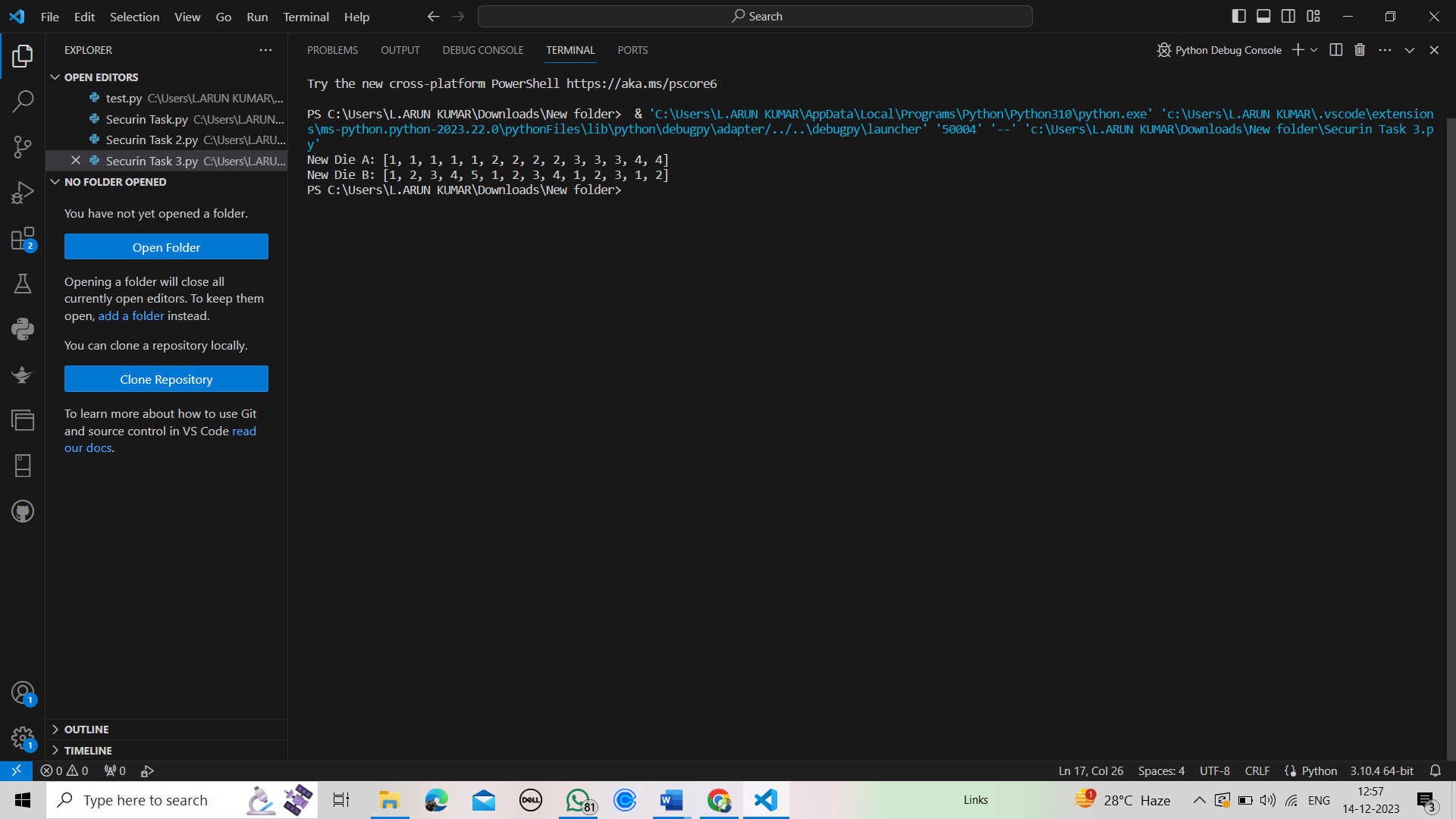


Fig. 3-b