**Part – B:**

**Code :**

**from itertools import combinations\_with\_replacement**

**from itertools import combinations**

**def find\_new\_die\_A\_and\_new\_die\_B(die\_a, die\_b):**

**probability = {2: 1, 12: 1, 3: 2, 11: 2, 4: 3, 10: 3, 5: 4, 9: 4, 6: 5, 8: 5, 7: 6}**

**unique\_number = []**

**all\_sums = []**

**for i in die\_a:**

**for j in die\_b:**

**sum\_of\_pairs = i + j**

**all\_sums.append(sum\_of\_pairs)**

**if(sum\_of\_pairs not in unique\_number):**

**unique\_number.append(sum\_of\_pairs)**

**new\_probability = {}**

**for each in unique\_number:**

**count = all\_sums.count(each)**

**# Add key and value to new\_probability dictionary**

**new\_probability[each] = count**

**if(new\_probability == probability):**

**print("New\_die\_A : ", die\_a)**

**print("New\_die\_B : ", die\_b)**

**# all combinations of die\_A (i.e, New\_Die A[x] <= 4)**

**die\_A = list(combinations\_with\_replacement([1, 2, 3, 4], 6))**

**# all combinations of die\_B**

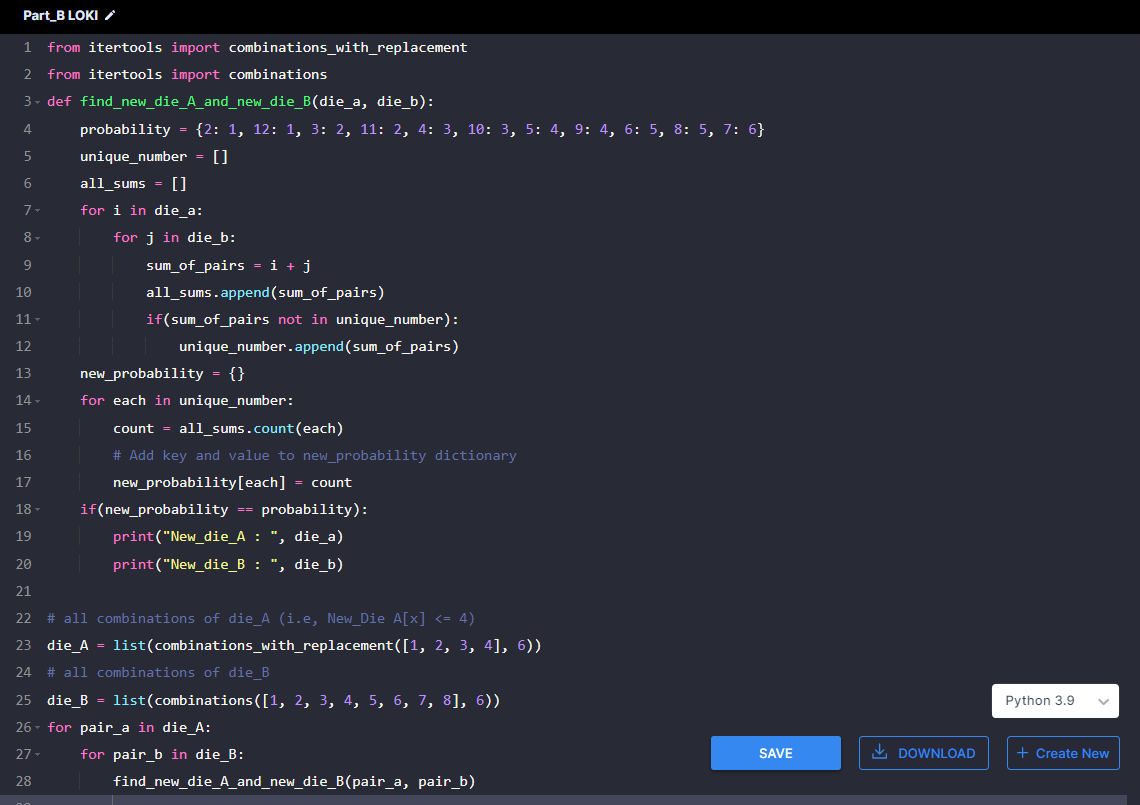
**die\_B = list(combinations([1, 2, 3, 4, 5, 6, 7, 8], 6))**

**for pair\_a in die\_A:**

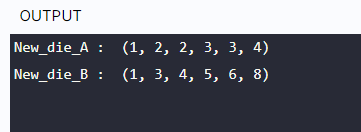
**for pair\_b in die\_B:**

**find\_new\_die\_A\_and\_new\_die\_B(pair\_a, pair\_b)**

**Code Screenshot :**

****

**Output :**

****

**Explanation :**

* Create a list die\_A of all combinations based on the given condition. No value greater than 4 (New\_Die A[x] <= 4).
* Similarly, die\_B of all combinations based on the given condition. (i.e. even more than 6).
* Two for loop were used for die\_A and die\_B.
* Define a function find\_new\_die\_A\_and\_new\_die\_B and pass each item of die\_A and die\_B as arguments.
* I define the probability of two dice values.

(i.e, probability = {2: 1, 12: 1, 3: 2, 11: 2, 4: 3, 10: 3, 5: 4, 9: 4, 6: 5, 8: 5, 7: 6})

* Two loop used for each pair, calculate the sum\_of\_pairs added to all\_sums list.
* If condition used sum\_of\_pairs value not in unique\_list, add the sum\_of\_pair value to unique\_list.
* Declare an empty dictionary (new\_probability = {})
* Using the count function, find the count of each sum\_value and add it to the dictionary key, value pair.
* Return new\_die\_A, new\_die\_B if the initial probability value is equal to the new\_probability value.