**-------------------------------------------PART – ‘A’------------------------------------------------**

**Question-1:**

**Code Explanation**:

* The variable total\_comb is calculated by multiplying the number of faces on Die A (6) with the number of faces on Die B (6).
* It represents the total number of combinations possible when rolling both Die A and Die B together.

**Question-2:**

**Code Explanation:**

**Printing All Possible Combinations:**

* The nested for loops iterate through each face of Die A and Die B.
* For each combination of faces (i, j), it prints the combination in the format (i, j).

**Combinations Distribution:**

* Another set of nested for loops is used to print the distribution of all possible combinations.
* Each row represents a different sum, and it prints the values contributing to each sum.

**Question -3:**

**Code Explanation:**

* For each possible sum (ranging from 2 to 12), it calculates the frequency of that sum using the getFreq method.
* The getFreq method returns the frequency of the sum based on its value.
* It then calculates the probability of each sum occurring by dividing its frequency by the total number of combinations.
* Finally, it prints the probability of each sum in the format

"P(Sum=x) = frequency/total\_com".

**getFreq Method**:

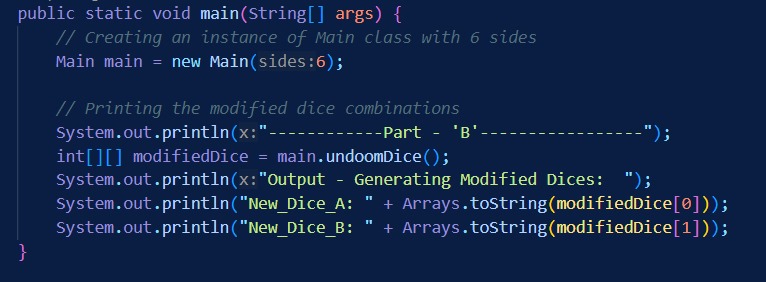
* This method takes a sum as input and returns its frequency.
* It handles cases where the sum is less than 2, greater than 12, or falls within the range of 2 to 7 differently, returning appropriate frequencies for each case.

**--------------------------------------------------PART - ‘B’------------------------------------------------**

**Generating Modified Dice Combinations**

**Code Explanation**:

* **Code Description:**The code consists of a single class named Main, which contains methods for generating modified dice combinations and testing their functionality.
* **Main Class:**The Main class serves as the entry point for the program. It includes a constructor for initializing the number of sides for the dice and methods for generating modified dice combinations.
* **Constructor:**The constructor Main(int sides) initializes the number of sides for the dice. It takes an integer parameter sides representing the number of sides for each die.
* **undoomDice() Method:**The undoomDice() method generates all possible combinations of dice values, sorts them based on the sum of their values, and modifies them based on a specific condition. It returns a 2D array containing the modified dice combinations.
* **main Method:**The main method creates an instance of the Main class with 6 sides, calls the undoomDice() method to get the modified dice combinations, and prints the output.
* **Variable and Data Structures:**The code utilizes lists, arrays, and integer variables to store and manipulate dice values and combinations.
* **Code Flow:**The flow of the code involves generating all possible combinations of dice values, sorting them, and then modifying them based on a specific condition.
* **Example:**Below is an example of how the code can be used to generate modified dice combinations:



* **Conclusion**:The provided code efficiently generates modified dice combinations, demonstrating its usefulness in various applications.