

Throw Those Dice

In this problem you will complete the `ThrowThoseDice` class simulating rolling a collection of dice with varying number of faces and analyzes the results. You will need to implement the constructor and four methods.

Note: A six-sided die, that is used in many games. The numbers 1 – 6 are written on each side of the die. Similarly, on an n-sided dice, the numbers 1 – n are written on each side of the die.

The first task is to implement the default constructor (has no parameters), and two helper methods. The constructor creates the class and creates a data structure for storing the collection of dice that will be rolled. (The collection contains zero dice at this time.) The first helper method is `clearDice()`, which clears all dice from the collection (The collection now has zero dice). The second helper method is `addDice(int numDices)`, which adds a die with `numSides` to the existing collection of dice. These methods are not tested. You may implement these methods anyway you wish as they are only used to add and subtract dice to the collection of dice.

The first tested method is the `getNumTimes(int target)`, which returns the number of different ways to roll a total of `target` with the current collection of dice. For example:

- Given one five sided dice, there is only one way to roll a 4 (or any other number between one and 5)
- Given two six sided dice, there are 6 ways to roll a 7
- Given three six sided dice, there are 15 ways to roll a 7

The following code shows the results of the `getNumTimes` method.

| The following code | Returns |
|---|---------|
| <code>ThrowThoseDice ttd = new ThrowThoseDice();</code> <code>ttd.addDice(5);</code> | |
| <code>ttd.getNumTimes(4);</code> | 1 |
| <code>ttd.clearDice();</code> <code>ttd.addDice(6);</code> <code>ttd.addDice(6);</code> | |
| <code>ttd.getNumTimes(7);</code> | 6 |
| <code>ttd.addDice(6);</code> | |
| <code>ttd.getNumTimes(7);</code> | 15 |

The second tested method is `getOdds(int target)`, which returns the odds (in non simplified fractional form) of rolling sum equal to `target` with the current collection of dice. This method should return a `int[2]` with the numerator being stored in index 0, and the denominator stored in index 1.

To simplify the problem, you should not reduce the fractions. The denominator should be equal to the product of the number of faces of all the dice in the collection.

For example:

- Given one five sided dice, the odds of rolling a 4 is $1/5$
- Given two six sided dice, the odds of rolling a 7 is $6/(6*6)$ or $6/36$
- Given three six sided dice, the odds of rolling a 7 is $15/(6*6*6)$ or $15/216$
- Given three six sided dice and a four sided dice, the odds of rolling a 10 is $71/(6*6*6*4)$ or $71/864$

The following code shows the results of the `getOdds` method.

| The following code | Returns |
|---|-----------|
| <code>ThrowThoseDice ttd = new ThrowThoseDice();</code> <code>ttd.addDice(5);</code> | |
| <code>ttd.getOdds(4)[0]</code> | 1 |
| <code>ttd.getOdds(4)[1]</code> | 5 |
| <code>ttd.clearDice();</code> <code>ttd.addDice(6);</code> <code>ttd.addDice(6);</code> | |
| <code>ttd.getOdds(7)[0]</code> | 6 |
| <code>ttd.getOdds(7)[1]</code> | 36 |
| <code>ttd.addDice(6);</code> | |
| <code>ttd.getOdds(7)[0]</code> | 15 |
| <code>ttd.getOdds(7)[1]</code> | $6*6*6$ |
| <code>ttd.addDice(4);</code> | |
| <code>ttd.getOdds(10)[0]</code> | 71 |
| <code>ttd.getOdds(10)[1]</code> | $6*6*6*4$ |