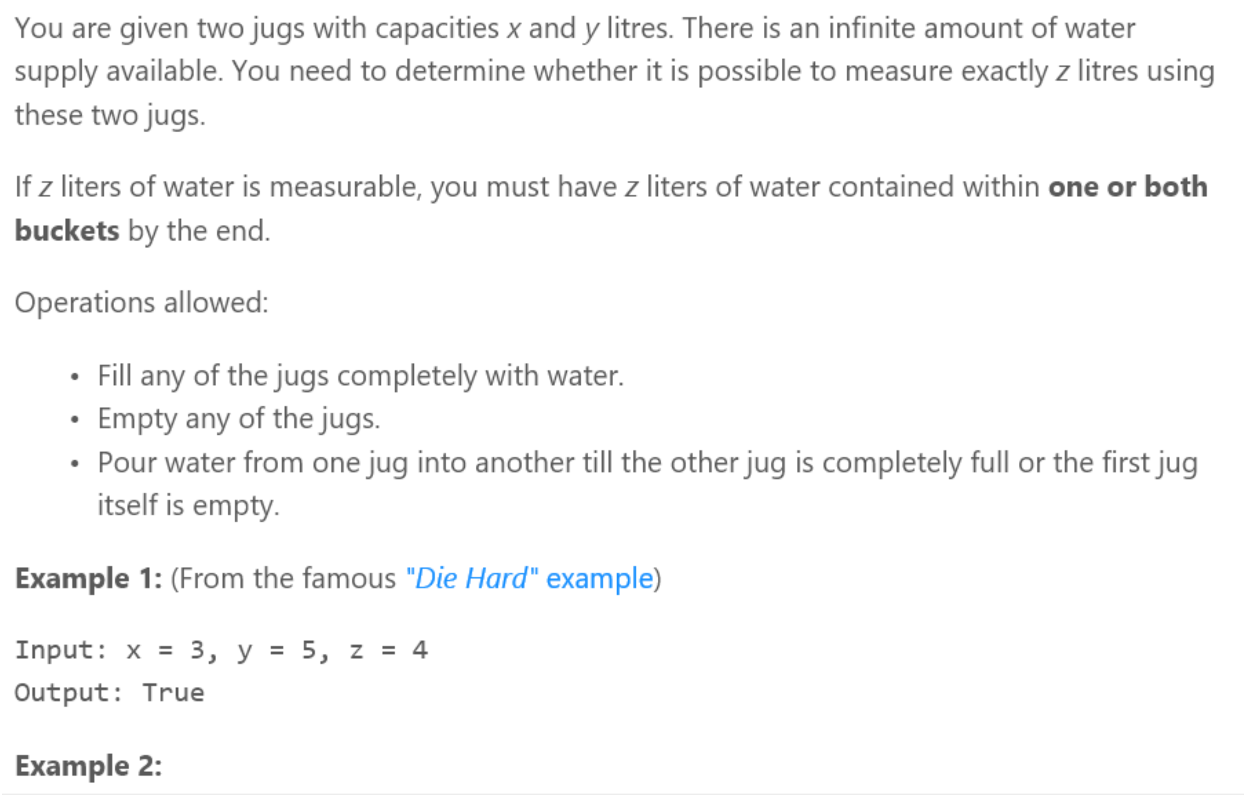


You are given two jugs with capacities *x* and *y* litres. There is an infinite amount of water supply available. You need to determine whether it is possible to measure exactly *z* litres using these two jugs.

If *z* liters of water is measurable, you must have *z* liters of water contained within **one or both buckets** by the end.

Operations allowed:

- Fill any of the jugs completely with water.
- Empty any of the jugs.
- Pour water from one jug into another till the other jug is completely full or the first jug itself is empty.

Example 1: (From the famous "*Die Hard*" example)

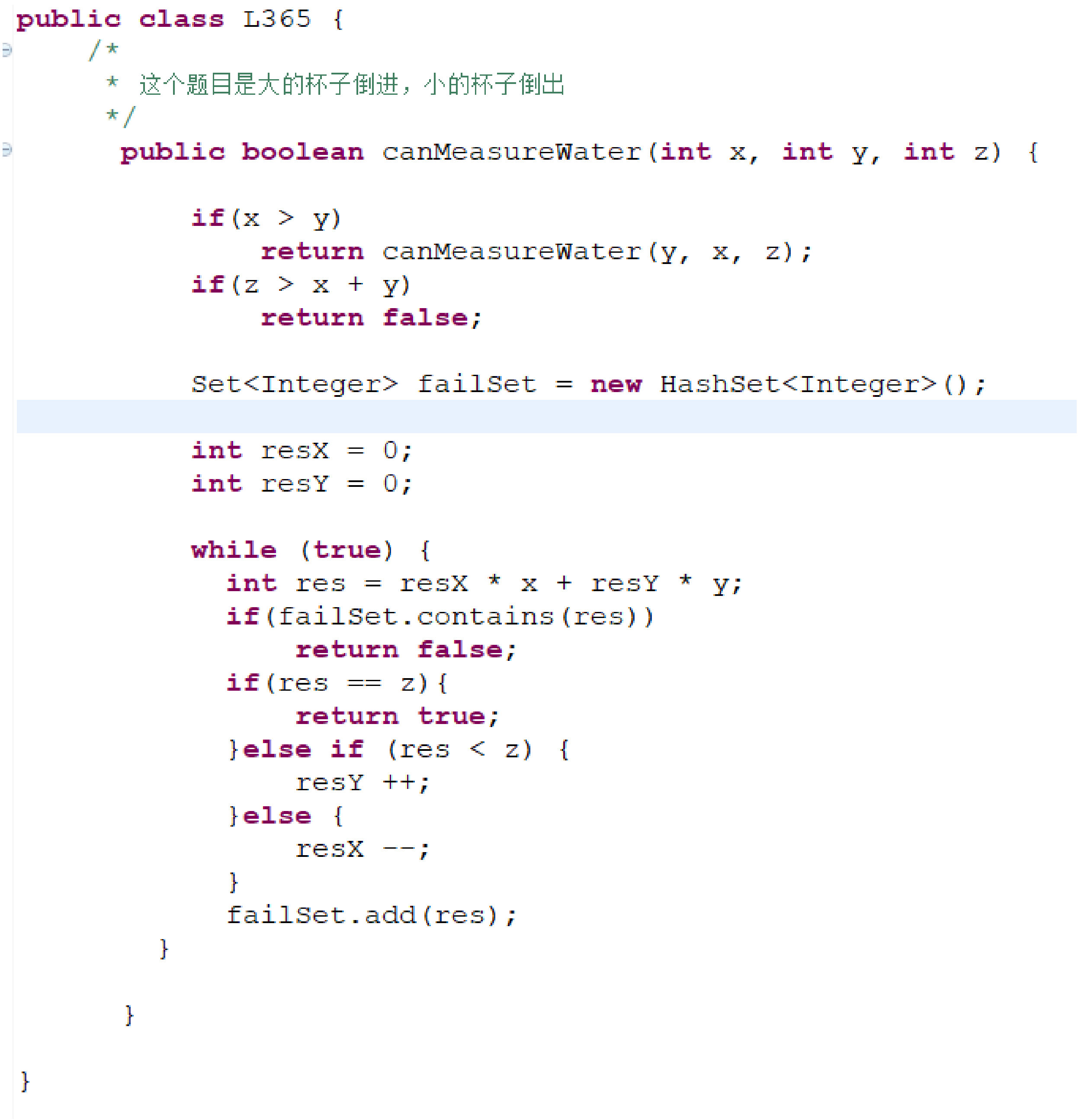
```
Input: x = 3, y = 5, z = 4
Output: True
```

Example 2:

这道题意思是有两个杯子，容量分别为x和y，问我们通过用两个杯子往里倒水，和往出舀水，问能不能使容器中的水刚好为z升。那么我们可以用一个公示来表达。

z = m ＊ x + n ＊ y

其中m，n为舀水和倒水的次数，正数表示往里舀水，负数表示往外倒水，那么题目中的例子可以写成4 = （-2） ＊ 3 + 2 ＊ 5，即3升的水罐往外倒了两次水，5升的水罐往里舀了两次水，那么问题就变成了对于任意给定的x，y，z，存不存在m和n使得上面的等式成立。



```
public class L365 {
    /*
     * 这个题目是大的杯子倒进，小的杯子倒出
     */
      public boolean canMeasureWater(int x, int y, int z) {

          if(x > y)
              return canMeasureWater(y, x, z);
          if(z > x + y)
              return false;

          Set<Integer> failSet = new HashSet<Integer>();

          int resX = 0;
          int resY = 0;

          while (true) {
            int res = resX * x + resY * y;
            if(failSet.contains(res))
                return false;
            if(res == z){
                return true;
            }else if (res < z) {
                resY ++;
            }else {
                resX --;
            }
            failSet.add(res);
        }

      }

}
```