```
/*
排排坐，分糖果。

我们买了一些糖果 candies，打算把它们分给排好队的 n = num_people 个小朋友。

给第一个小朋友 1 颗糖果，第二个小朋友 2 颗，依此类推，直到给最后一个小朋友 n 颗糖果。

然后，我们再回到队伍的起点，给第一个小朋友 n + 1 颗糖果，第二个小朋友 n + 2 颗，依此类推，直到给最后一个小朋友 2 * n 颗糖果。

重复上述过程（每次都比上一次多给出一颗糖果，当到达队伍终点后再次从队伍起点开始），直到我们分完所有的糖果。注意，就算我们手中的剩下糖果数不够（不比前一次发出的糖果多），这些糖果也会全部发给当前的小朋友。

返回一个长度为 num_people、元素之和为 candies 的数组，以表示糖果的最终分发情况（即 ans[i] 表示第 i 个小朋友分到的糖果数）。



示例 1：

输入：candies = 7, num_people = 4
输出：[1,2,3,1]
解释：
第一次，ans[0] += 1，数组变为 [1,0,0,0]。
第二次，ans[1] += 2，数组变为 [1,2,0,0]。
第三次，ans[2] += 3，数组变为 [1,2,3,0]。
第四次，ans[3] += 1（因为此时只剩下 1 颗糖果），最终数组变为 [1,2,3,1]。

示例 2：

输入：candies = 10, num_people = 3
输出：[5,2,3]
解释：
第一次，ans[0] += 1，数组变为 [1,0,0]。
第二次，ans[1] += 2，数组变为 [1,2,0]。
第三次，ans[2] += 3，数组变为 [1,2,3]。
第四次，ans[0] += 4，最终数组变为 [5,2,3]。



提示：

    1 <= candies <= 10^9
    1 <= num_people <= 1000

通过次数7,229
提交次数11,624

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/distribute-candies-to-people

*/
```

分析：

- 方法一：模拟分配过程即可.

方法一：C++_模拟分配过程

```cpp
class Solution
{
    public:
        vector<int> distributeCandies(int candies, int num_people)
        {
            vector<int> vi_ret(num_people,0);
            int count = 0;

            while(candies>0)
            {
                if(candies > count)
                {
                    vi_ret[count%num_people] += (count+1);
                    count++;
                    candies = candies - count;
                }
                else
                {
                    vi_ret[count%num_people] += candies;
                    candies -= candies;
                    break;
                }
            }

            return vi_ret;

        }
};

/*
27 / 27 个通过测试用例
  状态：通过
执行用时：0 ms
内存消耗：8.7 MB
提交时间：5 分钟之前
*/
```

AlimyBreak
2020.03.05(2020年第11周)