

和为 s 的连续正数序列

输入一个正整数 $target$ ，输出所有和为 $target$ 的连续正整数序列（至少含有两个数）。

序列内的数字由小到大排列，不同序列按照首个数字从小到大排列。

示例 1：

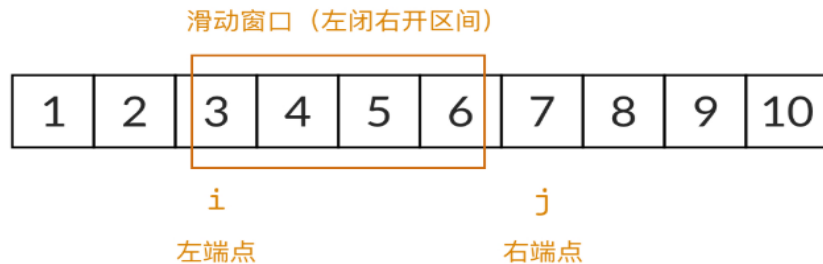
输入：target = 9

输出：[[2,3,4],[4,5]]

示例 2：

输入：target = 15

输出：[[1,2,3,4,5],[4,5,6],[7,8]]



当窗口的和小于 $target$ 的时候，窗口的和需要增加，所以要扩大窗口，窗口的右边界向右移动

当窗口的和大于 $target$ 的时候，窗口的和需要减少，所以要缩小窗口，窗口的左边界向右移动

当窗口的和恰好等于 $target$ 的时候，我们需要记录此时的结果。设此时的窗口为 $[i, j)$ ，那么我们已经找到了一个 i 开头的序列，也是唯一一个 i 开头的序列，接下来需要找 $i+1$ 开头的序列，所以窗口的左边界要向右移动



```

class Solution {
public:
    vector<vector<int>> findContinuousSequence(int target) {
        vector<vector<int>> res;
        int i=1; //左边界
        int j=1; //右边界
        int sum=0;
        while(i<=target/2)
        {
            if(sum>target) //滑动窗口需要缩小 但只能在右移的前提下 左边界右移且
            值缩小
            {
                sum=sum-i;
                i++;
            }
            else if(sum<target) //滑动窗口需要扩大 但只能在右移的前提下 右边界
            右移且值增大
            {
                sum=sum+j;
                j++;
            }
            else
            {
                vector<int> tmp;
                for(int k=i; k<j; k++)
                {
                    tmp.push_back(k); //一维序列 记录下和为 s 的序列
                }
                res.push_back(tmp); //记录下 所有 和为 s 的序列
                //滑动窗口整体右移滑动 右移左边界
                sum=sum-i;
                i++;
            }
        }
        return res;
    }
};

```