```
/*
找出所有相加之和为 n 的 k 个数的组合。组合中只允许含有 1 - 9 的正整数，并且每种组合中不存在重复的数字。

说明：

    所有数字都是正整数。
    解集不能包含重复的组合。

示例 1:

输入: k = 3, n = 7
输出: [[1,2,4]]

示例 2:

输入: k = 3, n = 9
输出: [[1,2,6], [1,3,5], [2,3,4]]

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/combination-sum-iii
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析:

- 可选不重复的可能性有$[1,9]$共计9个可选对象，我们可以使用9个二进制位分别表示某个数被选或未被选中;
- 遍历$i \in [0,511]$获取其中1的个数,若为k,则判断位对应的数之和是否为n.

方法一:C++_bit位表示法

```
class Solution
{
    private:
        int bit1Count(unsigned int n)
        {
            int count = 0;
            while (n)
            {
                count++;
                n &= (n - 1);
            }
            return count;
        }
    public:
        vector<vector<int>> combinationSum3(int k, int n)
        {
            unsigned int    i = 0;
            int             j = 0;
```

```
            vector<int>     temp;
            vector<vector<int>> ret_val;
            int sum = 0;
            for (i = 0; i < 512; i++)
            {
                if (bit1Count(i) == k)
                {
                    temp.clear();
                    sum = 0;
                    for (j = 0; j < 9; j++)
                    {
                        if (i & (0x01 << j))
                        {
                            temp.push_back(j + 1);
                            sum += (j + 1);
                        }
                    }
                    if (sum == n)
                    {
                        ret_val.push_back(temp);
                    }
                }
            }
            return ret_val;
        }
};

/*
执行结果：
通过
显示详情
执行用时 :0 ms, 在所有 C++ 提交中击败了100.00% 的用户
内存消耗 :8.4 MB, 在所有 C++ 提交中击败了92.65%的用户
*/
```

AlimyBreak
2019.10.09