

给定一个正整数 n ，生成一个包含 1 到 n^2 所有元素，且元素按顺时针顺序螺旋排列的正方形矩阵。

示例:

输入: 3

输出:

```
[
  [ 1, 2, 3 ],
  [ 8, 9, 4 ],
  [ 7, 6, 5 ]
]
```

思路：直接新建2维数组，数组值为0，然后按照[54. 螺旋矩阵.note](#)的遍历思路进行遍历，然后当前和前一位相加为当前位的值即可得到答案。

代码:

```
class Solution {
public:
    vector<vector<int>> generateMatrix(int n) {
        vector<vector<int>> result(n, vector<int>(n, 1));
        int x=0, y=1;
        int i=0;
        int bj=0;
        int num=1;
        while (num<n*n) {
            if (i==0) {
                result[x][y]+=result[x][y-1];
                if (y+1==result[0].size()-bj) {
                    x++;
                    i=1;
                } else {
                    y++;
                }
            } else if (i==1) {
                result[x][y]+=result[x-1][y];
                if (x+1==result.size()-bj) {
                    y--;
                    i=2;
                } else {
                    x++;
                }
            } else if (i==2) {
                result[x][y]+=result[x][y+1];
                if (y==0+bj) {
```

```
        x--;
        i=3;
        bj++;
    }else{
        y--;
    }
}else{
    result[x][y]+=result[x+1][y];
    if(x==0+bj){
        y++;
        i=0;
    }else{
        x--;
    }
}
num++;
}
return result;
}
};
```