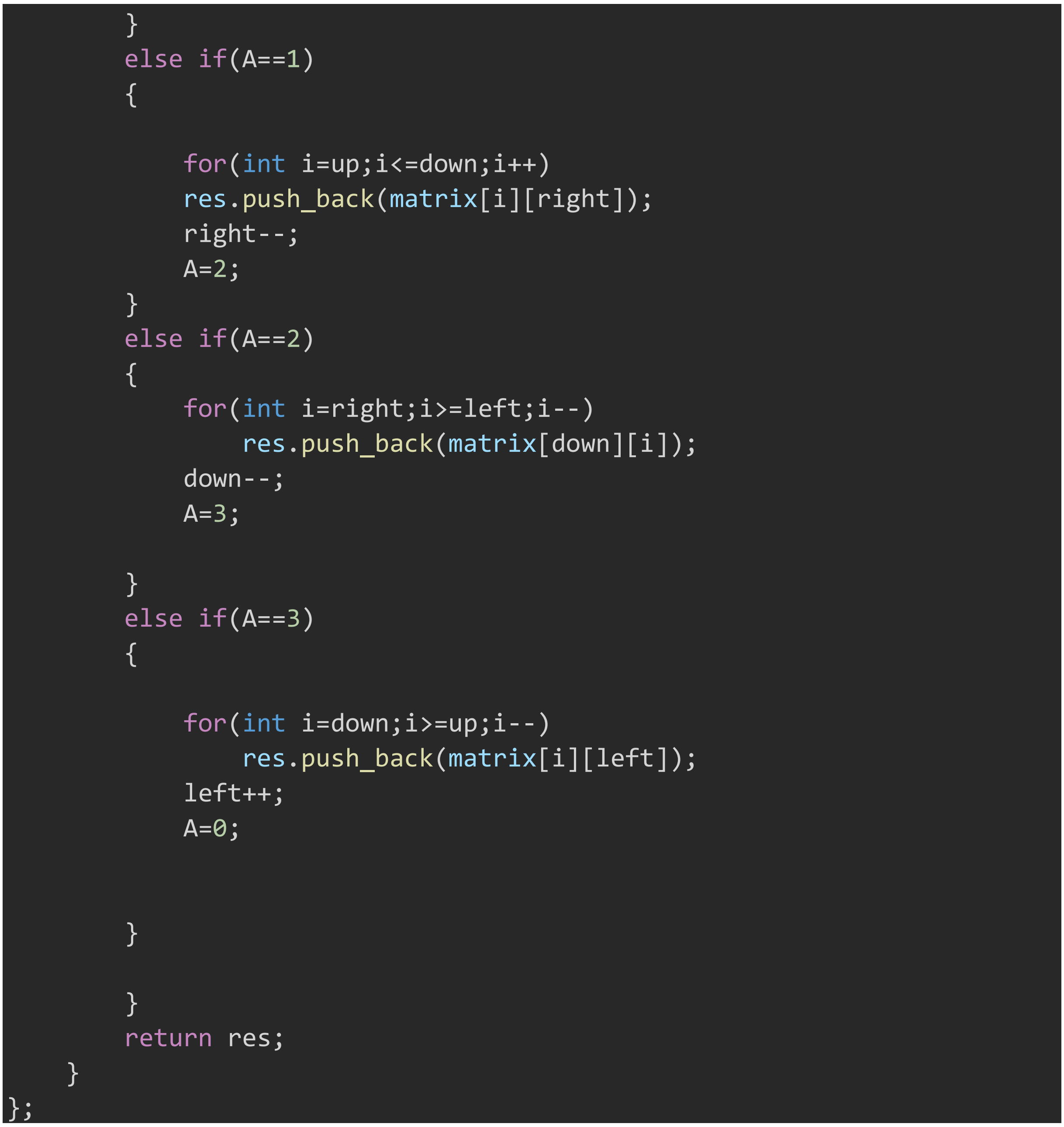
Subarray Sum Equals K

```
class Solution {
public:
    int subarraySum(vector<int>& nums, int k) {
        int count = 0;
        int prefixSum = 0;
        unordered_map<int, int> prefixSumCount;
        prefixSumCount[0] = 1;

        for (int i = 0; i < nums.size(); i++) {
            prefixSum += nums[i];


            if (prefixSumCount.count(prefixSum - k)) {
                count += prefixSumCount[prefixSum - k];
            }


            prefixSumCount[prefixSum]++;
        }

        return count;
    }
};
```

Spiral Matrix

```
class Solution {
public:
    vector<int> spiralOrder(vector<vector<int>>& matrix) {
        int n=matrix.size();
        int m=matrix[0].size();
        int A=0,up=0,down=n-1,left=0,right=m-1;
        vector<int>res;
        while(left<=right && up<=down)
        {
        if(A==0)
        {

            for(int i=left;i<=right;i++)
                res.push_back(matrix[up][i]);
            up++;
            A=1;
```



```
        }
        else if(A==1)
        {

            for(int i=up;i<=down;i++)
            res.push_back(matrix[i][right]);
            right--;
            A=2;
        }
        else if(A==2)
        {
            for(int i=right;i>=left;i--)
                res.push_back(matrix[down][i]);
            down--;
            A=3;

        }
        else if(A==3)
        {

            for(int i=down;i>=up;i--)
                res.push_back(matrix[i][left]);
            left++;
            A=0;


        }

        }
        return res;
    }
};
```