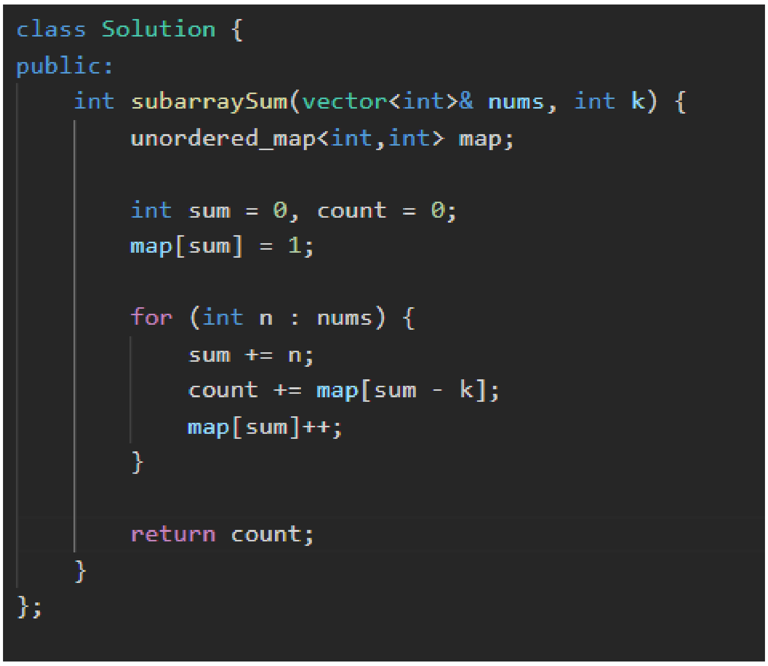
Spiral traversal

```
class Solution {
public:
    vector<int> spiralOrder(vector<vector<int>>& matrix) {
        int row = matrix.size();
        int column = matrix[0].size();
        int left = 0;int right = column-1;
        int top = 0;int bottom = row-1;
        vector<int> solution;
        while(top<=bottom && left<=right){
            for(int i=left;i<=right;i++){
                solution.push_back(matrix[top][i]);
            }
            top++;
            for(int i=top;i<=bottom;i++){
                solution.push_back(matrix[i][right]);
            }
            right--;
            if(top<=bottom){
                for(int i=right;i>=left;i--){
                solution.push_back(matrix[bottom][i]);
            }
            bottom--;
            }
            if(left<=right){
                for(int i=bottom;i>=top;i--){
                solution.push_back(matrix[i][left]);
            }
            left++;
            }

        }
        return solution;
    }
};
```

Subarray sum=k



```
class Solution {
public:
    int subarraySum(vector<int>& nums, int k) {
        unordered_map<int,int> map;

        int sum = 0, count = 0;
        map[sum] = 1;

        for (int n : nums) {
            sum += n;
            count += map[sum - k];
            map[sum]++;
        }

        return count;
    }
};
```