Task 3

DATE:8/01/24

SPIRAL MATRIX

```
def spiralOrder(matrix):
    ans = []


    if (len(matrix) == 0):
        return ans


    m = len(matrix)
    n = len(matrix[0])
    seen = [[0 for i in range(n)] for j in range(m)]
    dr = [0, 1, 0, -1]
    dc = [1, 0, -1, 0]
    x = 0
    y = 0
    di = 0
for i in range(m * n):
        ans.append(matrix[x][y])
        seen[x][y] = True
        cr = x + dr[di]
        cc = y + dc[di]


        if (0 <= cr and cr < m and 0 <= cc and cc < n and not(seen[cr][cc])):
            x = cr
            y = cc
        else:
            di = (di + 1) % 4
            x += dr[di]
            y += dc[di]
return ans
```

```
if __name__ == "__main__":
    a = [[1, 2,3],
        [4, 5, 6],
        [7, 8, 9]]



    for x in spiralOrder(a):
        print(x, end=" ")
    print()
```

2.SUBARRAY SUM == K

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

        //For fast I/O in C++
        ios_base::sync_with_stdio(false);
        cin.tie(NULL);

        int n = nums.size();
        if(n==0)
            return 0;

        unordered_map<int,int> mymap;   //Key = PrefixSUM, Value = Count of PrefixSUM.
        int currSUM = 0;
        int i = 0;
int count = 0;

        while(i<n)
        {
            currSUM += nums[i];

            if(currSUM == k)    //We found a new subArray with SUM = k
                count += 1;

            if(mymap.find(currSUM-k)!=mymap.end())
                count += mymap[currSUM-k];

            mymap[currSUM] += 1;
i += 1;
        }
        return count;
    }
};
```