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

1. 2. 3. 6. 9. 8. 7. 4. 5

n →

top = -1

right = 4

m . n .

for (i 0 to)

1. 2. 3. 4

m ↓ 5. 6. 7. 8

4 dirs

keep going until
reaching boundary.

9. 10. 11. 12

↑ 13. 14. 15. 16.

left = -1

bottom = 4.

←

```
const fn = (matrix) => {

  const m = matrix.length;

  const n = matrix[0].length;

  let top = 0;

  let right = m;

  let bottom = n;

  let left = -1;

  let i = 0;

  let j = 0;

  const output = [];
```

```
while(i < m && j < n) {
    while( j < right) {
        out put. push (matrix[i][j]);
        j += 1;
    }
    j -= 1;
    right -= 1;

    while( i < bottom) {
        out put. push( matrix[i][j]);
        i += 1;
    }
    i -= 1;
    bottom -= 1;

    while( j > left) {
        out put. push ( matrix[i][j]);
        j -= 1;
    }
    j += 1;
    left += 1;
```

```
    while (j > top) {
        output.push (matrix[i][j]);
        i -= 1;
    }
    i += 1;

    top += 1;
}

return output;
}
```