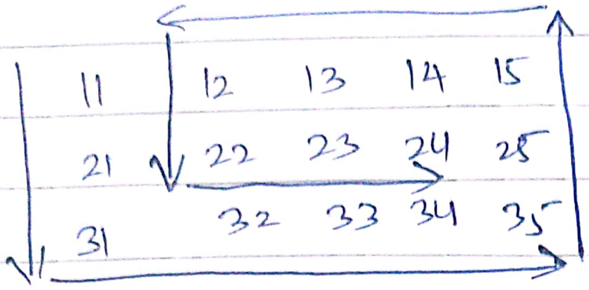
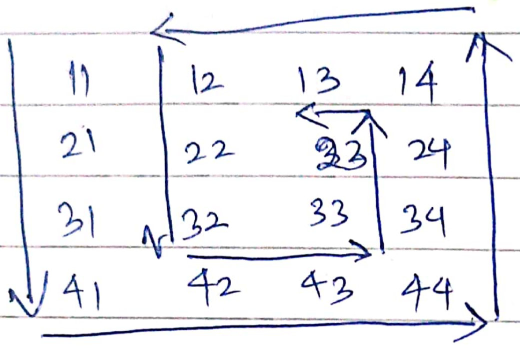
SPIRAL TRAVERSAL 3x5 Matrix Example



Another example 4x4



11 21 31 41 42 43 44
34 24 14 13 12 22 32
33 23
Output ⤴

Boxed Approach excluding Corners :: anti-clockwise

→ TO Identity box top-left and bottom-right indexes
Viz min-row | min-colu & max-row | max-column.

[Left Wall] ⇒ Column fix, Row changes [minrow to max col]

[Bottom Wall] ⇒ row fix, Column chges [min col to max Col]

[Right Wall] ⇒ row changes, Column fix [max row to minrow]

[Top Wall] ⇒ row, fix , column changes [max col to min Col]

Pseudo code

```
int minr = 0;
int minc = 0;
int maxr = arr.GetLength(0);
int maxc = arr.GetLength(1);
```
} initialization.

Boxed approach.

Top wall represents $\boxed{\text{min-row}}$

Left wall represents $\boxed{\text{min-column}}$

Right wall represents $\boxed{\text{Max-column}}$

11	12	13	14	15	16	17
21	22	23	24	25	26	27
31	32	33	34	35	36	37
41	42	43	44	45	46	47
51	52	53	54	55	56	57

Bottom wall represents $\boxed{\text{max-row}}$

After left wall → minc++
After bottom wall → maxr--
After right wall → maxc--
After top wall → minr--

```
int cnt = 0;
int tne = n×m [arr.GetLength(1) * arr.GetLength(0)]

while (cnt <tne) {
// Left Wall
for(i= minr, j=minc; i<=maxr && cnt<tne; i++){
    Console.WriteLine(arr[i,j]);
        Cnt++;
  }
  minc++;

// bottom Wall
for(int i=maxr, j=minc; j<=maxc && cnt<tne; j++){
    ~~ar~~ Console.WriteLine(arr[i,j]);
        Cnt++;
  }
  maxr--;

//right wall
for(int i= maxr, j=maxc; i>=minr && cnt<tne ; i--){
      Console.WriteLine(arr[i,j]);
          Cnt++;
    }
    maxc--;
// top - wall
for(int i=minr, j=maxC; j>=minC && cnt<tne; j--) {
        Console.WriteLine(arr[i,j]);
         Cnt++;
    }
   minr++;
}
```