



Competitive Programming

2D Arrays-Spiral



B.Bhuvaneswaran, AP (SG) / CSE



9791519152



bhuvaneswaran@rajalakshmi.edu.in



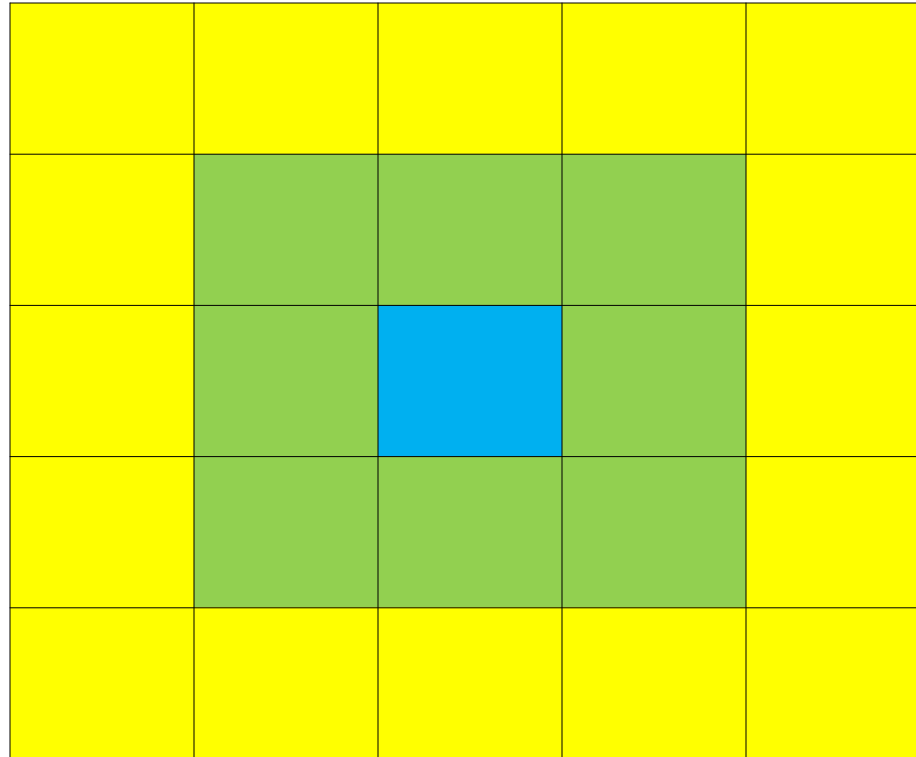
RAJALAKSHMI
ENGINEERING COLLEGE

An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

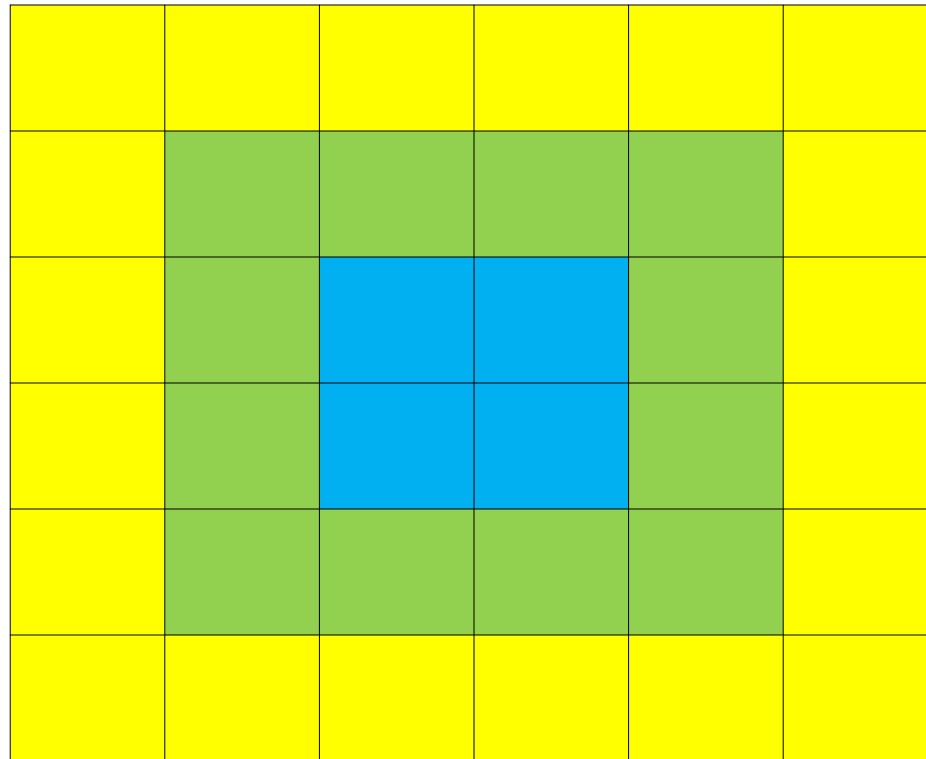
Spiral matrix for input 5

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

Approach for input 5



Approach for input 6



Hint

- For even numbers, the last square is always 2×2 .
- For odd numbers, the last square is always 1×1 .
- Number of full square = $N / 2$ (Even numbers)
- The last square (1×1) element for odd numbers is N^2 .

Approach

- Create an array ($N \times N$)
- Maintain a counter for the number
- Print each square separately ($N / 2$ squares)
 - Print top row
 - Print right column
 - Print bottom row
 - Print left column
 - If N is odd, print the last number

Pseudo code

input n

counter = 0

row = col = end_row = end_col = 0

for i = 0 to n / 2 - 1:

 row = col = i

 print top row

 print right column

 print bottom row

 print left column

If n % 2 == 1:

 print the last number

Pseudo code: Print top row

```
end_col = n - i - 1
```

```
while col < end_col:
```

```
    matrix[row][col] = ++counter
```

```
    col++
```


Pseudo code: Print right column

```
end_row = n - i - 1
```

```
while row < end_row:
```

```
    matrix[row][col] = ++counter
```

```
    row++
```

Pseudo code: Print bottom row

```
end_col = i
```

```
while col > end_col:
```

```
    matrix[row][col] = ++counter
```

```
    col--
```

Pseudo code: Print left column

```
end_row = i
```

```
while row > end_row:
```

```
    matrix[row][col] = ++counter
```

```
    row--
```

Pseudo code: Print last number (odd)

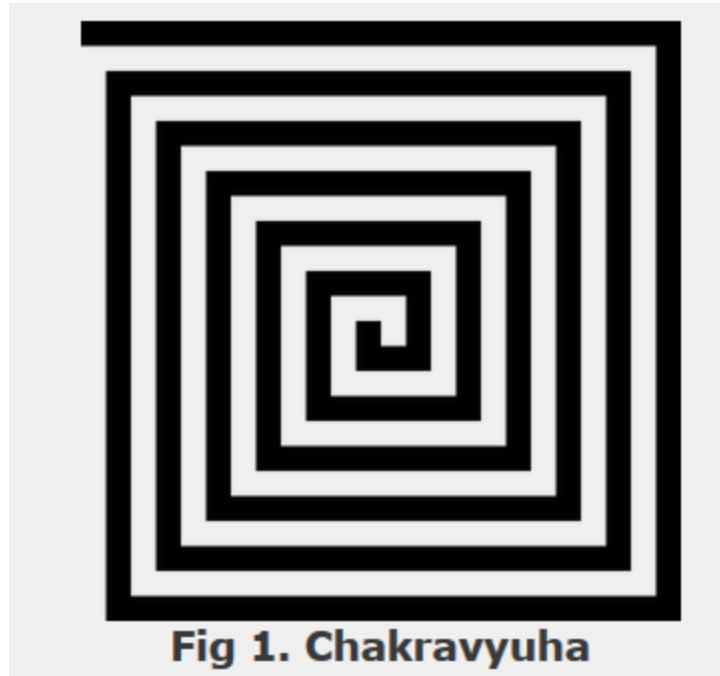
```
if n % 2 == 1
```

```
    matrix[n / 2][n / 2] = ++counter
```

Pseudo code: Print matrix

```
for i : 0 to n - 1
    for j : 0 to n - 1
        print matrix[i][j] + "\t"
    print new line
```

Chakravyuha



Chakravyuha

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

Fig 2. Army unit placements in Chakravyuha of size 5

Chakravyuha

Input

2

Output

1 2

4 3

Total Power points : 1

(0,0)

Input

5

Output

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

Total Power points : 3

(0,0)

(4,2)

(3,2)

Pseudo code

```
input n
powerpoints = 1 + (n * n / 11)
pp_list[2 * powerpoints]
pp_counter = 0
pp_list[pp_counter++] = 0
pp_list[pp_counter++] = 0
counter = 0
row = col = end_row = end_col = 0
for i = 0 to n / 2 - 1:
    row = col = i
    print top row
    print right column
    print bottom row
    print left column
If n % 2 == 1:
    print the last number
```

Pseudo code: Print top row

```
end_col = n - i - 1
```

```
while col < end_col:
```

```
    chakra[row][col] = ++counter
```

```
    if counter % 11 == 0:
```

```
        pp_list[pp_counter++] = row
```

```
        pp_list[pp_counter++] = col
```

```
    col++
```

Pseudo code: Print right column

```
end_row = n - i - 1
```

```
while row < end_row:
```

```
    chakra[row][col] = ++counter
```

```
    if counter % 11 == 0:
```

```
        pp_list[pp_counter++] = row
```

```
        pp_list[pp_counter++] = col
```

```
    row++
```

Pseudo code: Print bottom row

```
end_col = i
```

```
while col > end_col:
```

```
    chakra[row][col] = ++counter
```

```
    if counter % 11 == 0 :
```

```
        pp_list[pp_counter++] = row
```

```
        pp_list[pp_counter++] = col
```

```
    col--
```

Pseudo code: Print left column

```
end_row = i
```

```
while row > end_row:
```

```
    chakra[row][col] = ++counter
```

```
    if counter % 11 == 0:
```

```
        pp_list[pp_counter++] = row
```

```
        pp_list[pp_counter++] = col
```

```
    row--
```

Pseudo code: Print last number (odd)

```
if n % 2 == 1:
```

```
    chakra[n / 2][n / 2] = ++counter
```

```
    if counter % 11 == 0:
```

```
        pp_list[pp_counter++] = n / 2
```

```
        pp_list[pp_counter++] = n / 2
```

Queries?

Thank You...!