

Custom Search

COURSES

Login

HIRE WITH US



Print n x n spiral matrix using O(1) extra space

Given a number n, print a n x n spiral matrix (of numbers from 1 to n x n) in clockwise direction using O(1) space.

Example :

Input: n = 5

Output:

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

We strongly recommend you to minimize your browser and try this yourself first.

The solution becomes simple if extra space is allowed. We allocate memory for n x n matrix and for every element starting from n*n to 1, we start filling out matrix in spiral order. To maintain the spiral order four loops are used, each for top, right, bottom and left corner of the matrix.

Get closer to
the right people.

Learn more

LinkedIn Sales Navigator



But how to solve it in O(1) space?

An n x n matrix has $\text{ceil}(n/2)$ square cycles. A cycle is formed by ith row, (n-i+1)th column, (n-i+1)th row and ith column where i varies from 1 to $\text{ceil}(n/2)$.

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

1. The first cycle is formed by elements of its first row, last column, last row and first column (marked by red). The first cycle consists of elements from n*n to (n-2)*(n-2) + 1. i.e. from 25 to 10.
2. The second cycle is formed by elements of second row, second-last column, second-last row and second column (marked by blue). The second cycle consists of elements from (n-2)*(n-2) to (n-4)*(n-4) + 1. i.e. from 9 to 2.
3. The third cycle is formed by elements of third row, third-last column, third-last row and third column (marked by black). The third cycle consists of elements from (n-4)*(n-4) to 1. i.e. only 1.

The idea is for each square cycle, we associate a marker to it. For outer cycle, the marker will have value 0, for second cycle, it will have value 1 and for third cycle it has value 2. In general for a n x n matrix, i'th cycle will have marker value of $i - 1$.

If we divide the matrix into two parts, upper right triangle(marked by orange) and lower left triangle(marked by green), then using the marker x, we can easily calculate the value that will be present at the index(i, j) in any n x n spiral matrix using the below formula –

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

For upper right half,

$$\text{mat}[i][j] = (n-2*x)*(n-2*x) - (i-x) - (j-x)$$

For lower left half,

$$\text{mat}[i][j] = (n-2*x-2)*(n-2*x-2) + (i-x) + (j-x)$$

Below is the implementation of the idea.

C++

```
// C++ program to print a n x n spiral matrix
// in clockwise direction using O(1) space
#include <bits/stdc++.h>
using namespace std;

// Prints spiral matrix of size n x n containing
// numbers from 1 to n x n
void printSpiral(int n)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            // x stores the layer in which (i, j)th
            // element lies
            int x;

            // Finds minimum of four inputs
            x = min(min(i, j), min(n-1-i, n-1-j));

            // For upper right half
            if (i <= j)
                printf("%d\t ", (n-2*x)*(n-2*x) - (i-x) - (j-x));

            // for lower left half
            else
                printf("%d\t ", (n-2*x-2)*(n-2*x-2) + (i-x) + (j-x));

            printf("\n");
        }
    }
}

// Driver code
int main()
{
    int n = 5;

    // print a n x n spiral matrix in O(1) space
    printSpiral(n);

    return 0;
}
```

Java

```
// Java program to print a n x n spiral matrix
// in clockwise direction using O(1) space
```

```

import java.io.*;
import java.util.*;

class GFG {

    // Prints spiral matrix of size n x n
    // containing numbers from 1 to n x n
    static void printSpiral(int n)
    {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {

                // x stores the layer in which (i, j)th
                // element lies
                int x;

                // Finds minimum of four inputs
                x = Math.min(Math.min(i, j),
                    Math.min(n - 1 - i, n - 1 - j));

                // For upper right half
                if (i <= j)
                    System.out.print((n - 2 * x) * (n - 2 * x) -
                        (i - x) - (j - x) + "\t");

                // for lower left half
                else
                    System.out.print((n - 2 * x - 2) * (n - 2 * x - 2) +
                        (i - x) + (j - x) + "\t");

            }
            System.out.println();
        }
    }

    // Driver code
    public static void main(String args[])
    {
        int n = 5;

        // print a n x n spiral matrix in O(1) space
        printSpiral(n);
    }
}

/*This code is contributed by Nikita Tiwari.*/

```

Python3

```

# Python3 program to print a n x n spiral matrix
# in clockwise direction using O(1) space

# Prints spiral matrix of size n x n
# containing numbers from 1 to n x n
def printSpiral(n) :

    for i in range(0, n) :
        for j in range(0, n) :

            # Finds minimum of four inputs
            x = min(min(i, j), min(n - 1 - i, n - 1 - j))

            # For upper right half
            if (i <= j) :
                print((n - 2 * x) * (n - 2 * x) -
                    (i - x) - (j - x), end = "\t")

            # For lower left half
            else :
                print(((n - 2 * x - 2) *
                    (n - 2 * x - 2) +
                    (i - x) + (j - x)), end = "\t")

            print()

# Driver code
n = 5

# print a n x n spiral matrix
# in O(1) space

```

```
printSpiral(n)
```

```
# This code is contributed by Nikita Tiwari.
```

C#

```
// C# program to print a n x n
// spiral matrix in clockwise
// direction using O(1) space
using System;

class GFG {

    // Prints spiral matrix of
    // size n x n containing
    // numbers from 1 to n x n
    static void printSpiral(int n)
    {
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {

                // x stores the layer in which
                // (i, j)th element lies
                int x;

                // Finds minimum of four inputs
                x = Math.Min(Math.Min(i, j),
                    Math.Min(n - 1 - i, n - 1 - j));

                // For upper right half
                if (i <= j)
                    Console.Write((n - 2 * x) *
                        (n - 2 * x) -
                        (i - x) - (j - x) + "\t");

                // for lower left half
                else
                    Console.Write((n - 2 * x - 2) *
                        (n - 2 * x - 2) +
                        (i - x) + (j - x) + "\t");

            }
            Console.WriteLine();
        }
    }

    // Driver code
    public static void Main()
    {
        int n = 5;

        // print a n x n spiral
        // matrix in O(1) space
        printSpiral(n);
    }
}

// This code is contributed by KRV
```

PHP

```
<?php
// PHP program to print a n x n
// spiral matrix in clockwise
// direction using O(1) space

// Prints spiral matrix of size
// n x n containing numbers
// from 1 to n x n
function printSpiral($n)
{
    for ($i = 0; $i < $n; $i++)
    {
        for ($j = 0; $j < $n; $j++)
        {
```

```

// x stores the layer in
// which (i, j)th element lies
$x;

// Finds minimum of four inputs
$x = min(min($i, $j), min($n - 1 - $i,
                        $n - 1 - $j));

// For upper right half
if ($i <= $j)
    echo "\t ", ($n - 2 * $x) *
                ($n - 2 * $x) -
                ($i - $x) - ($j - $x);

// for lower left half
else
    echo "\t ", ($n - 2 * $x - 2) *
                ($n - 2 * $x - 2) +
                ($i - $x) + ($j - $x);
}
echo "\n";
}
}

// Driver code
$n = 5;

// print a n x n spiral
// matrix in O(1) space
printSpiral($n);

// This code is contributed by ajit
?>

```

Output :

```

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

```

Exercise

For a given number n, print a n x n spiral matrix in counter clockwise direction using O(1) space.

This article is contributed by **Aditya Goel**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

**Recommended Posts:**

[Rotate a matrix by 90 degree without using any extra space | Set 2](#)

[Rotate a matrix by 90 degree in clockwise direction without using any extra space](#)

[Print a given matrix in spiral form](#)

[Print a given matrix in reverse spiral form](#)

[Print K'th element in spiral form of matrix](#)

[Print a matrix in a spiral form starting from a point](#)

Print a given matrix in spiral form using direction tracking method

Print a given matrix in counter-clock wise spiral form

Circular Matrix (Construct a matrix with numbers 1 to m*n in spiral way)

Find the sum of the diagonal elements of the given N X N spiral matrix

Sum of both diagonals of a spiral odd-order square matrix

Program to print Spiral Pattern

Inplace (Fixed space) M x N size matrix transpose | Updated

Printing Matrix Chain Multiplication (A Space Optimized Solution)

Print the matrix diagonally downwards

Improved By : KRV, jit_t



Deliciously Easy Caesar Salad

Crush your 2019 diet goals with this healthy Caesar Salad chopped kit by Fresh Express

Fresh Express

VISIT SITE

Article Tags : [Matrix](#) [pattern-printing](#) [spiral](#)

Practice Tags : [pattern-printing](#) [Matrix](#)



5

4.4

☐ To-do ☐ Done

Based on 34 vote(s)

Feedback/ Suggest Improvement

Add Notes

Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

[About Us](#)
[Careers](#)
[Privacy Policy](#)
[Contact Us](#)

PRACTICE

[Courses](#)
[Company-wise](#)
[Topic-wise](#)
[How to begin?](#)

LEARN

[Algorithms](#)
[Data Structures](#)
[Languages](#)
[CS Subjects](#)
[Video Tutorials](#)

CONTRIBUTE

[Write an Article](#)
[Write Interview Experience](#)
[Internships](#)
[Videos](#)

@geeksforgeeks, Some rights reserved