Pattern Problems: Steps to Solve

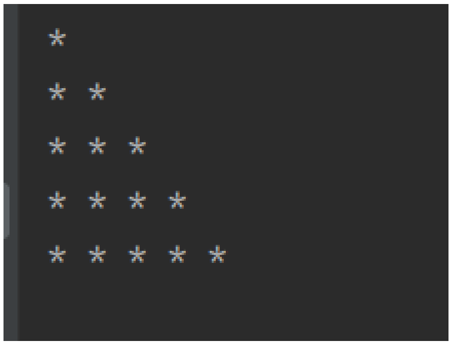
1. **Find how many times the outer loop will run.**
   Number of Lines = Number of Rows
   => Number of Times Outer Loop will Run

2. **Identify For Every Row number,**
- How many Columns are there?
- Types of Elements in Column

3. **What Do you Need to print?**
- if number needs to be printed, start loop with row = 1

in general: always try to find relationship between no of rows, no of columns and no spaces etc.

Problem 1: Print Below Pattern



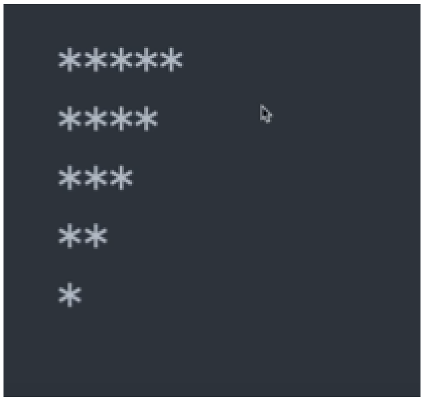
```
*
* *
* * *
* * * *
* * * * *
```

Solution :

```
import java.util.Scanner;

public class Pattern1 {
   public static void main (String [] args)
   {
      Scanner sc = new Scanner (System.in);
      int n1 = sc.nextInt();
      pattern1(n1);
   }
   static void pattern1(int n)
   {
      for (int row = 1; row <= n; row++)
      {
         //for every row, find no of column
         for (int col = 1; col <=row; col++)
         {
            // for every column, what need to be printed
            System.out.print("* ").
         }
         System.out.println();
      }
```

```
         }
      }
```

Problem : Print Below Pattern

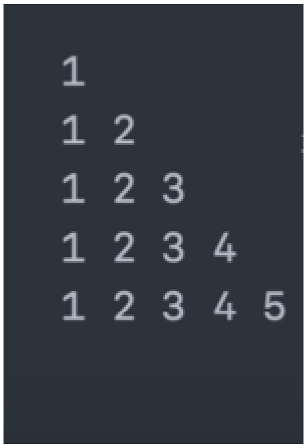


```
*****
****
***
**
*
```

Solution:

```
 static void pattern3(int n)
   {
      //n=5
      for ( int row = 1; row <= n; row++ )
      {
         //for every row ,find no of column
         for (int col = 1; col <= n-row+1 ; col++)
         {
            // for every column, what need to be printed
            System.out.print("* ");
         }
         System.out.println();
      }
   }
```

Poblem : Print Below Pattern

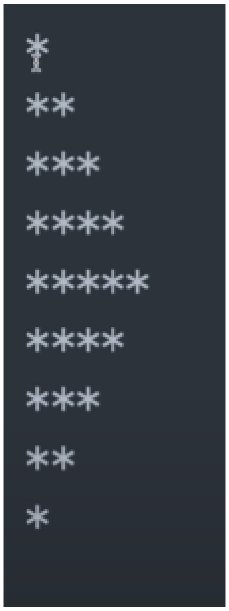


```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Solution:

```
static void pattern4(int n)
   {
     //n=5
     for ( int row = 1; row <= n; row++ )
     {
       //for every row ,find no of column
       for (int col = 1; col <= row ; col++)
       {
         // for every column, what need to be printed
         System.out.print(col +" ");
       }
       System.out.println();
     }
   }
```

Problem : Print Below Pattern



```
*
**
***
****
*****
****
***
**
*
```

Hint : Find relation between row, n and column ,when row < =n and row >n

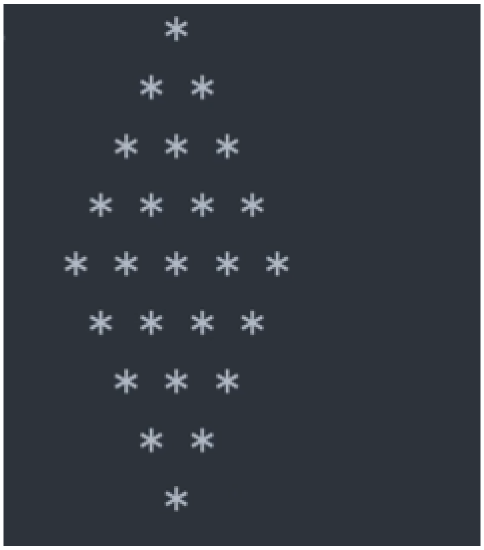
Solution :

total no of lines = 2n-1

```
  static void pattern5(int n)
   {
     //n=5
     int noOfColumn;
     for ( int row_no = 1; row_no < 2*n; row_no++ )
     {
       //for every row_no ,find no of column
       noOfColumn = (row_no <=n) ? row_no : (2*n - row_no);
```

```
    for (int col = 1; col <= noOfColumn ; col++)
    {
      // for every column, what need to be printed
      System.out.print("* ");
    }
    System.out.println();
  }
}
```

Problem : Print Below Pattern

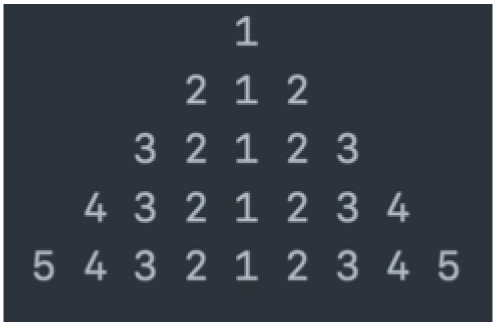


```
    * 
   * * 
  * * * 
 * * * * 
* * * * * 
 * * * * 
  * * * 
   * * 
    * 
```

Solution:
Hint : Find Relation among n, row, no of spaces , no of column.

```
static void pattern28(int n)
{
  //n=5
  int noOfColumn;
  for ( int row_no = 1; row_no < 2*n; row_no++ )
  {
    //for every row_no ,find no of column
    noOfColumn = (row_no <=n) ? row_no : (2*n - row_no);
    // For every row, print no of spaces
    for (int space = 1; space <= n-noOfColumn ; space++) {
      System.out.print(" ");
    }
    for (int col = 1; col <= noOfColumn ; col++)
    {
      // for every column, what need to be printed
      System.out.print("* ");
    }
    System.out.println();
  }
}
```

Problem: Print Below Pattern

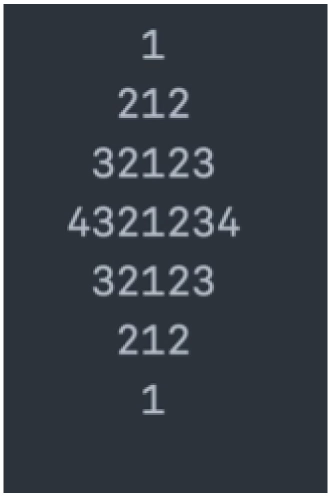


```
        1
      2 1 2
    3 2 1 2 3
  4 3 2 1 2 3 4
5 4 3 2 1 2 3 4 5
```

Solution:

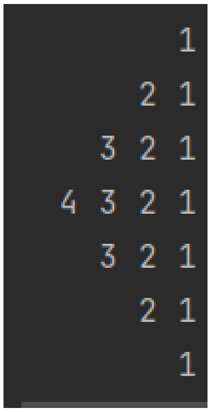
```
static void pattern29(int n)
 {
     //n=5
     for ( int row_no = 1; row_no <=n; row_no++ )
     {
        //for every row_no
        // For every row, print no of spaces
        for (int space = 1; space <= n-row_no ; space++)
       {
           //two spaces will be printed for each
           //since at each step ,by 1 extra space, first print is being shifted to right
           System.out.print("  ");
        }

        //print row_no to 1
        for (int r = row_no; r >=1;  r--)
        {
           // for every column, what need to be printed
           System.out.print(r+" ");
        }

        //print 2 to row_no
        for (int tr = 2; tr <=row_no; tr++)
        {
           System.out.print(tr+" ");
        }

        System.out.println();
     } // End Outer loop
}
```

Problem: Print Below Pattern



```
   1
  212
 32123
4321234
 32123
  212
   1
```

Solution:
Total n = 2n-1;
Hint : Focus on below part first

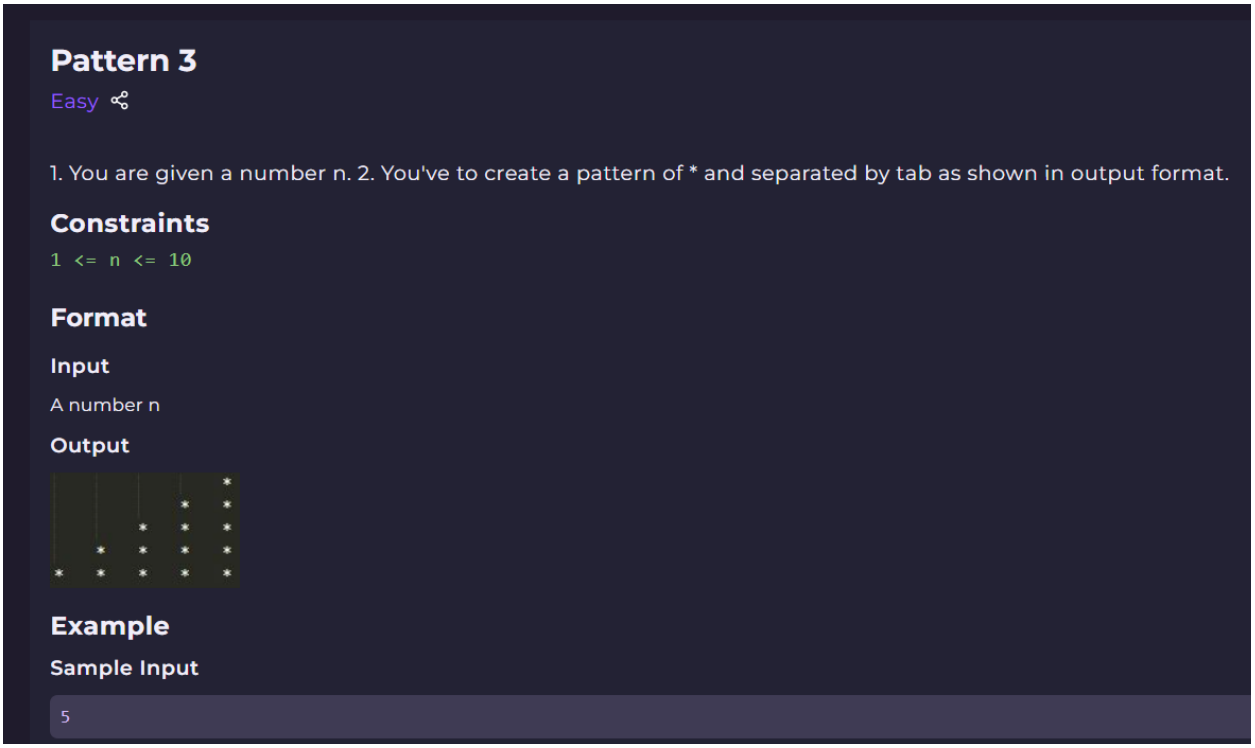


```
      1
    2 1
  3 2 1
4 3 2 1
  3 2 1
    2 1
      1
```

try to find relation between row_no and n ,derive no of columns for each half.

Code:

```
static void pattern17(int n)
  {
      //n=4

       // Outer Loop will run 2*n-1 times
      for ( int row_no = 1; row_no <=2*n-1; row_no++ )
      {
         //for every row_no, find no of columns
         int c = row_no > n ? 2*n - row_no : row_no;
         // For every row, print no of spaces
         for (int space = 0; space <= n-c ; space++) {
            //two spaces will be printed for each
            //since at each step ,by 1 extra space, first print is being shifted to right
            System.out.print("  ");
         }
         //print col to 1
         for (int col = c; col >=1;  col--)
         {
            // for every column, what need to be printed
            System.out.print(col+" ");
         }

         //print 2 to col
         for (int tc = 2; tc <=c; tc++)
         {
            System.out.print(tc+" ");
         }

         System.out.println();
      }

  }
```



Pattern 3

Easy

1. You are given a number n. 2. You've to create a pattern of * and separated by tab as shown in output format.

Constraints

```
1 <= n <= 10
```

Format

Input

A number n

Output

```
				*
			*	*
		*	*	*
	*	*	*	*
*	*	*	*	*
```

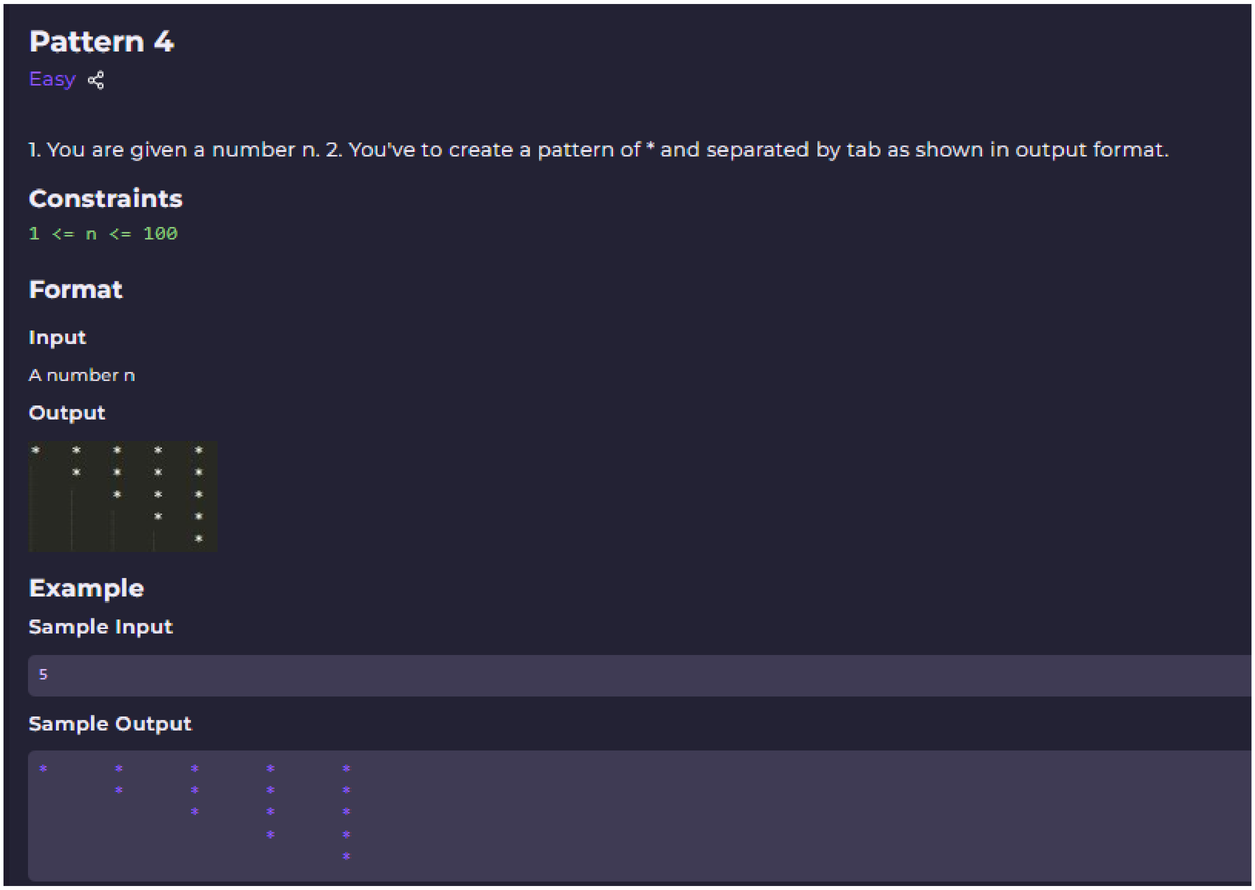
Example

Sample Input

```
5
```

```
import java.util.*;

public class Main {

  public static void main(String[] args)
  {
    Scanner scn = new Scanner(System.in);
```

```
     int n = scn.nextInt();

    for(int row =1; row <=n ; row++)
    {
      // for each row ,get the spaces, print it
      int totalSpaces = n-row;

      for(int space =1; space <= totalSpaces ; space++)
      {
        System.out.print("\t");

      }
      // for each row ,print the column
      for(int col =1; col <= row ; col++)
      {
        System.out.print("*\t");

      }
      System.out.println();
    }// end of loop
  }
}
```



Pattern 4

Easy

1. You are given a number n. 2. You've to create a pattern of * and separated by tab as shown in output format.

Constraints

```
1 <= n <= 100
```

Format

Input

A number n

Output

```
*	*	*	*	*
	*	*	*	*
		*	*	*
			*	*
				*
```

Example

Sample Input

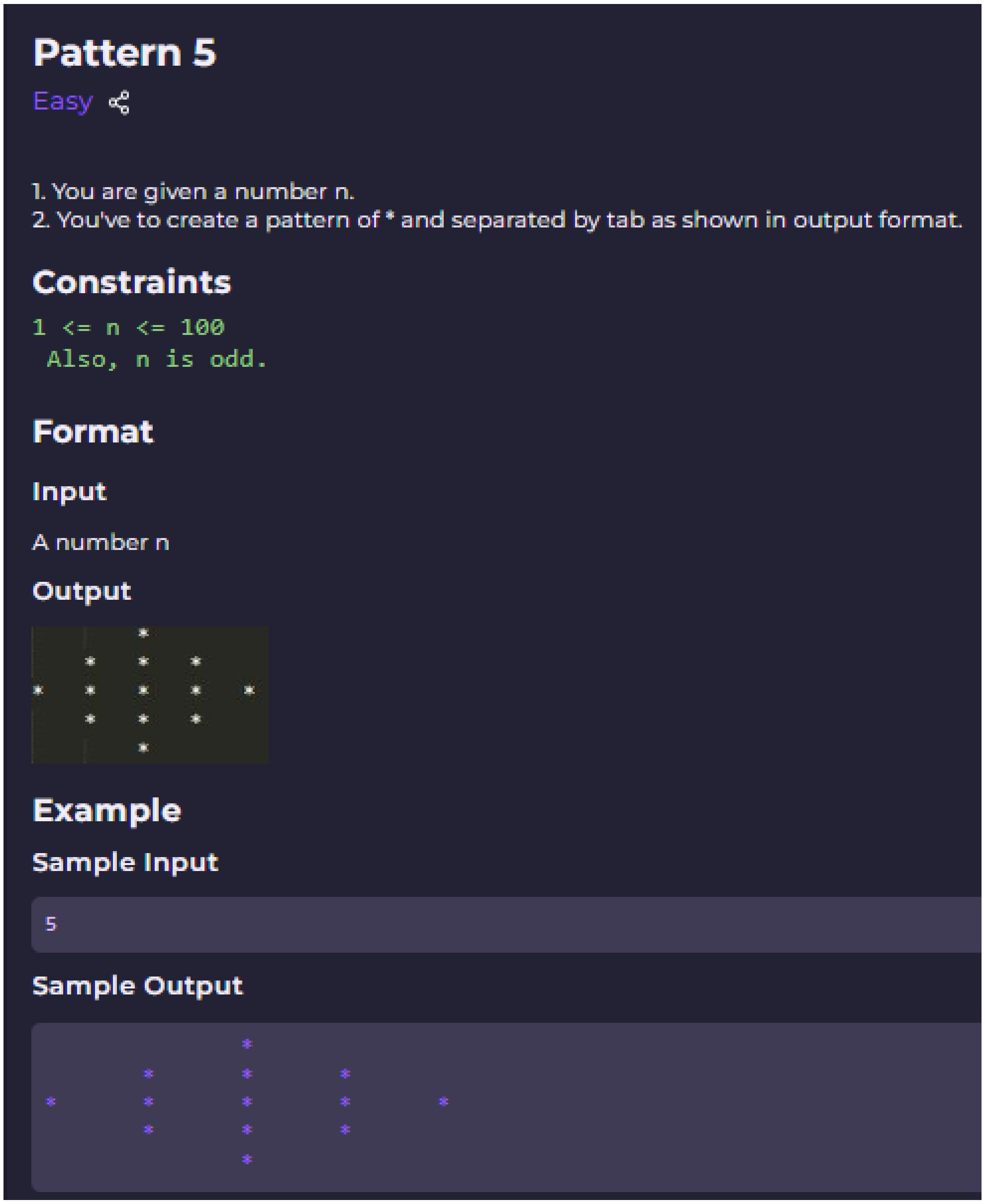
```
5
```

Sample Output

```
*	*	*	*	*
	*	*	*	*
		*	*	*
			*	*
				*
```

```
import java.util.*;
public class Main
{
    public static void main(String[] args)
    {
        Scanner scn = new Scanner(System.in);

        int n = scn.nextInt();

        for(int row =1; row <=n ; row++)
        {
            // for each row ,get the spaces
            int totalSpaces = row-1;
            for(int space =1; space <= totalSpaces ; space++)
            {
                System.out.print("\t");
            }
            // for each row ,get the no of column
            for(int col =1; col <= n-row+1 ; col++)
            {
                System.out.print("*\t");
            }
            System.out.println();
        }
    }
}
Time Complexitiy :
```

The time complexity of this solution is O(n^2), where n is the given input number since the outer for loop implements for n times and the inner loop for stars also implements for n times and when this loop has to execute only once then the loop for spaces has to implement for n - 1 times. So in terms of big ‘O’ notations, the complexity is O(n^2).



Pattern 5

Easy

1. You are given a number n.
2. You've to create a pattern of * and separated by tab as shown in output format.

Constraints

```
1 <= n <= 100
 Also, n is odd.
```

Format

Input

A number n

Output

```
        *
    *   *   *
*   *   *   *   *
    *   *   *
        *
```

Example

Sample Input

```
5
```

Sample Output

```
                *
        *       *       *
*       *       *       *       *
        *       *       *
                *
```

Solution:

1.try to find realtion between n ,spaces and no of columns

row _no	spaces	no_of_col(stars)
1	2	1
2	1	3
3	0	5
4	1	3
5	2	1

Code 1:

```
public static void pattern5(int n)
   {
      int col =1;
      int sp = n/2;
      for (int row_no = 1; row_no <=n ; row_no++)
      {
         // For each row
         // First, print spaces
         for (int spaces = 1; spaces <=sp ; spaces++) {
            System.out.print("\t");

         }
         // print the column
         for (int col_no = 1; col_no <=col ; col_no++) {
            // print the content
            System.out.print("*\t");

         }
         //find No of spaces and columns
         if(row_no <=n/2)
         {
            //spaces are decreasing by 1
            sp--;
            // columns are increasing by 2
            col = col+2;
         }
         else
         {
            //spaces are increasing by 1
            sp++;
            //col are decreasing by 2
            col =col -2;
         }
```

```
        // change the line
        System.out.println();
      }
    }
```

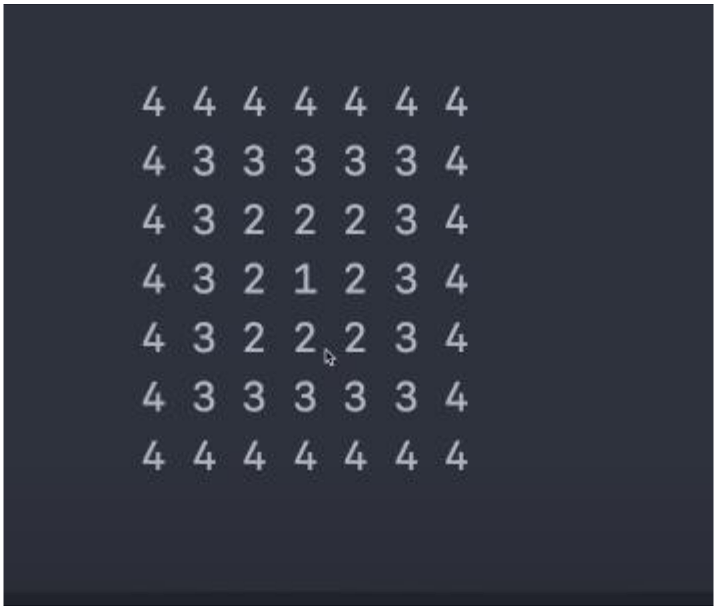
Code 2:

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        // write ur code here

         int n = scn.nextInt();
         int n1=2;
        int qutiont = n/2; e.g 5/2 =2 .

        int constant_one =1;

        int totalTabs =constant_one;

        int total_col = constant_one;

        int partDecider= qutiont+constant_one;

        // 1. Total no of rows is n
        for(int row =1; row <=n ; row++)
        {

            if( row <= partDecider)
            {

               // 2. total no of tabs
                totalTabs = qutiont -row+1;

                // 3. print the tabs
               for(int tab =1; tab <= totalTabs ; tab++)
                {
                System.out.print("\t");
                }

                // 4. for each row ,get the no of column

                for(int col =1; col <= total_col; col++)
               {
                System.out.print("*\t");
               }
                 total_col = total_col+2;

               System.out.println();
            }
            // when row > partDecider
            else
            {
                     // total no of tabs
             totalTabs = constant_one;

              // 3. print the tabs
            for(int tab =1; tab <= totalTabs ; tab++)
            {
                System.out.print("\t");

            }
            totalTabs = constant_one++;

              // for each row ,get the no of column
                  total_col = n-n1;
                 for(int col =1; col <= total_col; col++)
                 {
                System.out.print("*\t");
                 }
            n1=n1+2;
            System.out.println();

            }


        }



    }
}
```

Time Complexity: O(n)

We are running a loop from 1 to n, so the time complexity becomes O(n).

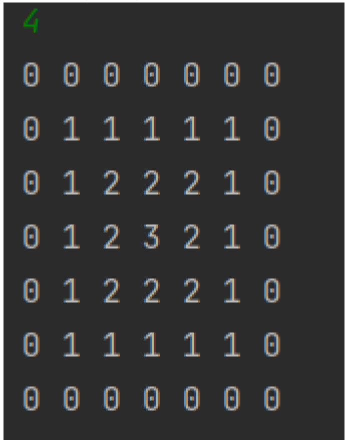
Problem: Print Below Pattern



```
4 4 4 4 4 4 4
4 3 3 3 3 3 4
4 3 2 2 2 3 4
4 3 2 1 2 3 4
4 3 2 2 2 3 4
4 3 3 3 3 3 4
4 4 4 4 4 4 4
```

Solution:

First try to print this:

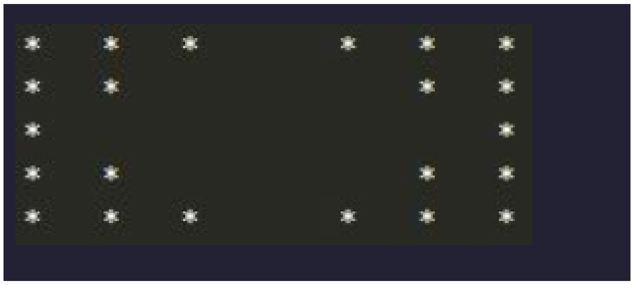


```
4
0 0 0 0 0 0 0
0 1 1 1 1 1 0
0 1 2 2 2 1 0
0 1 2 3 2 1 0
0 1 2 2 2 1 0
0 1 1 1 1 1 0
0 0 0 0 0 0 0
```

Every Cell Value at any index = min Distance from {left wall, bottom wall,right wall,upper wall}
=min {col , n-row, n-col,row}

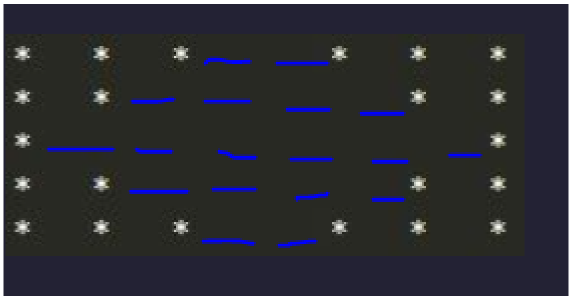
```
static void pattern31(int n)
  {
     int originalN =n;
     n = 2*n-2;
     for ( int row = 0; row <= n; row++ )
     {
        //for every row ,find no of column
        for (int col = 0; col <=n ; col++)
        {
            // for every column, what need to be printed

            int atEveryIndex = originalN-Math.min(Math.min(row,col),Math.min(n-row,n-col));

            //int atEveryIndex = Math.min(Math.min(row,col),Math.min(n-row,n-col));

            System.out.print(atEveryIndex +" ");
        }
        System.out.println();
      }
    }
```

Problem: Print Below pattern



```
*   *   *       *   *   *
*   *               *   *
*                       *
*   *               *   *
*   *   *       *   *   *
```

Solution:
Due to tab/space being printed at the last of star, one spaces would be less always.



```
*   *   *       *   *   *
*   *               *   *
*                       *
*   *               *   *
*   *   *       *   *   *
```

```
public static void pattern6(int n)
  {
    int col =n/2+1;
    int sp = 1;
    for (int row_no = 1; row_no <=n ; row_no++)
    {
      // For each row
      // print the column
      for (int col_no = 1; col_no <=col ; col_no++) {
        // print the content
        System.out.print("*\t");
      }
      //  print spaces
      for (int spaces = 1; spaces <=sp ; spaces++) {
        System.out.print("\t");

      }
      // print the column
      for (int col_no = 1; col_no <=col ; col_no++) {
        // print the content
        System.out.print("*\t");

      }
      //find spaces and columns
      if(row_no <=n/2)
      {
        //spaces are increasing by 2
        sp += 2;
        // columns are decreasing by 1
        col--;
      }
      else
      {
        //spaces are decreasing 2
        sp -= 2;
        //col are increasing by 1
        col++;
      }
      // change the line
      System.out.println();
    }
  }
```

Time Complexity: O(n^2)