

```

package com.kunal;

public class Main {
    public static void main(String[] args) {
        pattern31(4);
    }

    static void pattern31(int n) {
        int originalN = n;
        n = 2 * n;
        for (int row = 0; row <= n; row++) {
            for (int col = 0; col <= n; col++) {
                int atEveryIndex = originalN - Math.min(Math.min(row,
col), Math.min(n - row, n - col));
                System.out.print(atEveryIndex + " ");
            }
            System.out.println();
        }
    }

    static void pattern30(int n) {
        for (int row = 1; row <= n; row++) {

            for (int space = 0; space < n-row; space++) {
                System.out.print(" ");
            }

            for (int col = row; col >= 1; col--) {
                System.out.print(col + " ");
            }
            for (int col = 2; col <= row; col++) {
                System.out.print(col + " ");
            }

            System.out.println();
        }
    }

    static void pattern17(int n) {
        for (int row = 1; row <= 2 * n; row++) {

            int c = row > n ? 2 * n - row : row;

            for (int space = 0; space < n-c; space++) {
                System.out.print(" ");
            }

            for (int col = c; col >= 1; col--) {
                System.out.print(col + " ");
            }
            for (int col = 2; col <= c; col++) {
                System.out.print(col + " ");
            }
        }
    }
}

```

```

        }

        System.out.println();
    }
}

static void pattern28(int n) {
    for (int row = 0; row < 2 * n; row++) {
        int totalColsInRow = row > n ? 2 * n - row: row;

        int noOfSpaces = n - totalColsInRow;
        for (int s = 0; s < noOfSpaces; s++) {
            System.out.print(" ");
        }

        for (int col = 0; col < totalColsInRow; col++) {
            System.out.print("* ");
        }
        System.out.println();
    }
}

static void pattern5(int n) {
    for (int row = 0; row < 2 * n; row++) {
        int totalColsInRow = row > n ? 2 * n - row: row;
        for (int col = 0; col < totalColsInRow; col++) {
            System.out.print("* ");
        }
        System.out.println();
    }
}

static void pattern4(int n) {
    for (int row = 1; row <= n; row++) {
        // for every row, run the col
        for (int col = 1; col <= row; col++) {
            System.out.print(col + " ");
        }
        // when one row is printed, we need to add a newline
        System.out.println();
    }
}

static void pattern3(int n) {
    for (int row = 1; row <= n; row++) {
        // for every row, run the col
        for (int col = 1; col <= n-row+1; col++) {
            System.out.print("* ");
        }
        // when one row is printed, we need to add a newline
        System.out.println();
    }
}

```

```
}

static void pattern1(int n) {
    for (int row = 1; row <= n; row++) {
        // for every row, run the col
        for (int col = 1; col <= n; col++) {
            System.out.print("* ");
        }
        // when one row is printed, we need to add a newline
        System.out.println();
    }
}

static void pattern2(int n) {
    for (int row = 1; row <= n; row++) {
        // for every row, run the col
        for (int col = 1; col <= row; col++) {
            System.out.print("* ");
        }
        // when one row is printed, we need to add a newline
        System.out.println();
    }
}

}
```