**Problem Name: Mirror-Zip Palindrome Matrix**

**Topic:** Matrices

**Tags:** 2D Array, Symmetry, Palindrome, Pattern Printing

**Language used:** Java

**Difficulty:** Hard

**Problem Statement:**

You are given an odd integer N (1 ≤ N ≤ 99). You need to generate an N x N matrix pattern

that exhibits a property known as the **Mirror-Zip Palindrome Matrix**. The matrix must

satisfy the following rules:

1. Each row must be a palindrome.
2. The entire matrix must be symmetric horizontally and vertically.
3. The central row and the central column must contain a zigzag alphabet pattern starting from 'A' outward and reaching 'Z' at the center.
4. Each layer (like a concentric square around the center) increases in character from outermost 'A' to inner letters, up to 'Z' at the center.

**Input Format:**

A single odd integer N (1 ≤ N ≤ 99)

**Output Format:**

Print the N x N matrix pattern line-by-line, with each character separated by a space.

**Constraints:**

* 1 ≤ N ≤ 99
* N is always odd

**Sample Input 1:**

7

**Sample Output 1:**

A A A D A A A

A B B C B B A

A B C B C B A

D C B Z B C D

A B C B C B A

A B B C B B A

A A A D A A A

**Explanation of Sample Input 1:**

* The center character is always 'Z'.
* Rows and columns mirror around the center.
* Each row is a palindrome.
* Layers form increasing characters from 'A' towards the center.
* The matrix is symmetric both horizontally and vertically.

**Brute Force Approach:**

We create a matrix and loop through each cell (i, j):

* Compute layer using min(min(i, j), min(N - 1 - i, N - 1 - j))
* Assign char = 'A' + layer.
* Ensure row symmetry using mirroring.
* Overwrite center cell with 'Z'.
* Overwrite middle row and column with alphabet zigzag centered at 'Z'.

**Code :**

public class MirrorZipOptimized {

    public static void printMirrorZipMatrix(int N) {

        if (N % 2 == 0 || N < 1 || N > 99) {

            System.out.println("N must be an odd integer between 1 and 99.");

            return;

        }

        int mid = N / 2;

        for (int i = 0; i < N; i++) {

            for (int j = 0; j < N; j++) {

                if (i == mid && j == mid) {

                    // Center element

                    System.out.print("Z ");

                } else {

                    // Mirror-Zip Palindrome logic

                    int minRow = Math.min(i, N - 1 - i);

                    int minCol = Math.min(j, N - 1 - j);

                    int layer = Math.min(minRow, minCol);

                    char ch;

                    if (i == mid) {

                        ch = (char) ('A' + Math.abs(j - mid));

                    } else if (j == mid) {

                        ch = (char) ('A' + Math.abs(i - mid));

                    } else {

                        ch = (char) ('A' + layer);

                    }

                    System.out.print(ch + " ");

                }

            }

            System.out.println();

        }

    }

    public static void main(String[] args) {

        int N = 7;

        printMirrorZipMatrix(N);

    }

}