```c
/// Final 2557-1: N [Method 2]
/// Main idea: use a 2D array so that we can do the job by printing the left,
///   the middle, and the right in that order. Then, print out the array as
///   the final answer.
#include <stdio.h>

int A[300][300];

int main() {
    int N;
    scanf("%d", &N);

    // Mark array elements as empty spaces.
    for(int row = 0; row < N; ++row) {
        for(int col = 0; col < N; ++col) {
            A[row][col] = -1;
        }
    }

    // Fill the left-hand side.
    int count = 1;
    for(int row = N - 1; row >= 0; --row) {
        A[row][0] = count;
        ++count;
    }

    // Fill the diagonal.
    for(int row = 1; row < N - 1; ++row) {
        A[row][row] = count;
        ++count;
    }

    // Fill the right-hand side.
    for(int row = N - 1; row >= 0; --row) {
        A[row][N-1] = count;
        ++count;
    }

    // Print the result
    for(int row = 0; row < N; ++row) {
        for(int col = 0; col < N; ++col) {
            if(A[row][col] > 0 && col != N - 1)
                printf("%d ", A[row][col] % 10);
            else if(A[row][col] > 0 && col == N - 1)
                printf("%d", A[row][col] % 10);
            else
                printf("  ");
        }
        printf("\n");
    }

    return 0;
}
```