

```

1  /// Final 2557-1: N [Method 2]
2  /// Main idea: use a 2D array so that we can do the job by printing the left,
3  /// the middle, and the right in that order. Then, print out the array as
4  /// the final answer.
5  #include <stdio.h>
6
7  int A[300][300];
8
9  int main() {
10     int N;
11     scanf("%d", &N);
12
13     // Mark array elements as empty spaces.
14     for(int row = 0; row < N; ++row) {
15         for(int col = 0; col < N; ++col) {
16             A[row][col] = -1;
17         }
18     }
19
20     // Fill the left-hand side.
21     int count = 1;
22     for(int row = N - 1; row >= 0; --row) {
23         A[row][0] = count;
24         ++count;
25     }
26
27     // Fill the diagonal.
28     for(int row = 1; row < N - 1; ++row) {
29         A[row][row] = count;
30         ++count;
31     }
32
33     // Fill the right-hand side.
34     for(int row = N - 1; row >= 0; --row) {
35         A[row][N-1] = count;
36         ++count;
37     }
38
39     // Print the result
40     for(int row = 0; row < N; ++row) {
41         for(int col = 0; col < N; ++col) {
42             if(A[row][col] > 0 && col != N - 1)
43                 printf("%d ", A[row][col] % 10);
44             else if(A[row][col] > 0 && col == N - 1)
45                 printf("%d", A[row][col] % 10);
46             else
47                 printf(" ");
48         }
49         printf("\n");
50     }
51
52     return 0;
53 }

```