Daily Challenge **Happy Coding from necse** SkillRack

Power of 2 Matrix

The program must accept an integer **N** as the input. The program must form an integer matrix of size **(2^N)*(2^N)**. The program must fill the matrix with the integers submatrix by submatrix of size **2*2** in the order (Top-left, Top-right, Bottom-left and Bottom-right) as shown in the Example Input/Output section.

Boundary Condition(s):
1 <= N <= 7

Input Format:
The first line contains N.

Output Format:
The first 2^N lines contain the integer matrix as shown in the Example Input/Output section.

Example Input/Output 1:
Input:
3

Output:
1 2 5 6 17 18 21 22
3 4 7 8 19 20 23 24
9 10 13 14 25 26 29 30
11 12 15 16 27 28 31 32
33 34 37 38 49 50 53 54
35 36 39 40 51 52 55 56
41 42 45 46 57 58 61 62
43 44 47 48 59 60 63 64

Explanation:
Here **N=3**, so the size of the matrix is **8*8** (2^3 * 2^3).
In the **8*8** matrix, the integer **0** represents the cell is **empty**.
2^1 = **2**, so fill the **2*2** matrix with the integers in the order Top-left, Top-right, Bottom-left and Bottom-right.
1 2 0 0 0 0 0 0
3 4 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
2^2 = 4, so fill the **4*4** matrix with the integers in the order Top-left, Top-right, Bottom-left and Bottom-right.
1 2 5 6 0 0 0 0
3 4 7 8 0 0 0 0
9 10 13 14 0 0 0 0
11 12 15 16 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
2^3 = 8, so fill the **8*8** matrix with the integers in the order Top-left, Top-right, Bottom-left and Bottom-right.
1 2 5 6 17 18 21 22
3 4 7 8 19 20 23 24
9 10 13 14 25 26 29 30
11 12 15 16 27 28 31 32
33 34 37 38 49 50 53 54
35 36 39 40 51 52 55 56
41 42 45 46 57 58 61 62
43 44 47 48 59 60 63 64

Example Input/Output 2:
Input:
2

Output:
1 2 5 6
3 4 7 8
9 10 13 14
11 12 15 16

Example Input/Output 3:
Input:
1

Output:
1 2
3 4

Max Execution Time Limit: 50 millisecs

Ambiance

Java (12.0)

```
1  import java.util.*;
2  public class Hello {
3
4      static int value = 1;
5      public static void main(String[] args) {
6
7          Scanner sc = new Scanner(System.in);
8          int n = sc.nextInt();
9
10         int size = (int)Math.pow(2,n);
11
12         int mat[][] = new int[size][size];
13         brokeMatrix(mat,0,0,size-1,size-1);
14
15         for(int i=0;i<size;i++){
16             for(int j=0;j<size;j++){
17                 System.out.print(mat[i][j]+" ");
18             }
19                 System.out.println("");
20         }
21
22     }
23     public static void brokeMatrix(int[][] mat,int startRow,int startCol,int
           endRow,int endCol){
24
25        // System.out.println(startRow+"-"+startCol+"="+endRow+"-"+endCol);
26         if(startRow==endRow && startCol==endCol){
27             mat[startRow][startCol]=value++;
28             return;
29         }
30
31         brokeMatrix(mat,startRow,startCol,(startRow+endRow)/2,(startCol
               +endCol)/2);
32
33         brokeMatrix(mat,startRow,((startCol+endCol)/2)+1,(endRow+startRow)/2
               ,endCol);
34
35         brokeMatrix(mat,((startRow+endRow)/2)+1,startCol,endRow,(startCol
               +endCol)/2);
36
37         brokeMatrix(mat,((startRow+endRow)/2)+1,((startCol+endCol)/2)+1
               ,endRow,endCol);
38
39     }
40 }
```

1912067@nec

Code did not pass the execution

Input:

2

Expected Output:

```
1 2 5 6
3 4 7 8
9 10 13 14
11 12 15 16
```

Your Program Output:

```
1 2 5 6 -
3 4 7 8 -
9 10 13 14 -
11 12 15 16 -
```

Save Run