Java-矩阵数组-给定一个仅包含 0 和 1 的 N*N 二维矩阵

题目描述：

给定一个仅包含 0 和 1 的 N*N 二维矩阵，请计算二维矩阵的最大值，计算规则如下：

1、 每行元素按下标顺序组成一个二进制数（下标越大越排在低位），二进制数的值就是该行的值。矩阵各行值之和为矩阵的值。

2、允许通过向左或向右整体循环移动每行元素来改变各元素在行中的位置。

比如：[1,0,1,1,1]向右整体循环移动 2 位变为[1,1,1,0,1]，二进制数为 11101，值为 29。

[1,0,1,1,1]向左整体循环移动 2 位变为[1,1,1,1,0]，二进制数为 11110，值为 30。

输入描述：

1、输入的第一行为正整数，记录了 N 的大小，0 < N <= 20。

2、输入的第 2 到 N+1 行为二维矩阵信息，行内元素边角逗号分隔。

输出描述：
矩阵的最大值。
补充说明：

示例 1

输入：

5

1,0,0,0,1

0,0,0,1,1

0,1,0,1,0

1,0,0,1,1

1,0,1,0,1

输出：

122

说明：

第一行向右整体循环移动 1 位，得到本行的最大值[1,1,0,0,0]，二进制值为 11000，十进制值为 24。

第二行向右整体循环移动 2 位，得到本行的最大值[1,1,0,0,0]，二进制值为 11000，十进制值为 24。

第三行向左整体循环移动 1 位，得到本行的最大值[1,0,1,0,0]，二进制值为 10100，十进制值为 20。

第四行向右整体循环移动 2 位，得到本行的最大值[1,1,1,0,0]，二进制值为 11100，十进制值为 28。

第五行向右整体循环移动 1 位，得到本行的最大值[1,1,0,1,0]，二进制值为 11010，十进制值为 26。

因此，矩阵的最大值为 122。

```
import java.util.LinkedHashMap;

import java.util.Map;

import java.util.Scanner;


public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        sc.nextLine();

        int[][] arr = new int[n][n];
```

```
        for (int i = 0; i < n; i++) {
            String line = sc.nextLine();
            String[] numStrs = line.split(",");
            for (int j = 0; j < n; j++) {
                arr[i][j] = Integer.parseInt(numStrs[j]);
            }
        }
        int sum = 0;
        for (int i = 0; i < n; i++) {
            int maxValue = 0;
            for (int j = 0; j < n; j++) {
                int value = 0;
                for (int k = j; k < j + n; k++) {
                    value = value * 2 + arr[i][k % n];
                }
                if (value > maxValue) {
                    maxValue = value;
                }
            }
            sum += maxValue;
        }
        System.out.println(sum);
```

```
    }
}
```