

题目描述

小华是个对数字很敏感的小朋友，他觉得数字的不同排列方式有特殊美感。

某天，小华突发奇想，如果数字多行排列，第一行1个数，第二行2个，第三行3个，即第n行有n个数字，并且奇数行正序排列，偶数行逆序排列，数字依次累加。

这样排列的数字一定很有意思。聪明的你能编写代码帮助小华完成这个想法吗？

规则总结如下：

- a、每个数字占据4个位置，不足四位用*补位，如1打印为1***。
- b、数字之间相邻4空格。
- c、数字的打印顺序按照正序逆序交替打印，奇数行正序，偶数行逆序。
- d、最后一行数字顶格，第n-1行相对第n行缩进四个空格

输入描述

第一行输入为N，表示打印多少行; 1<=N<=30

输入：2

输出描述

xxxX1***

3***xxxX2***

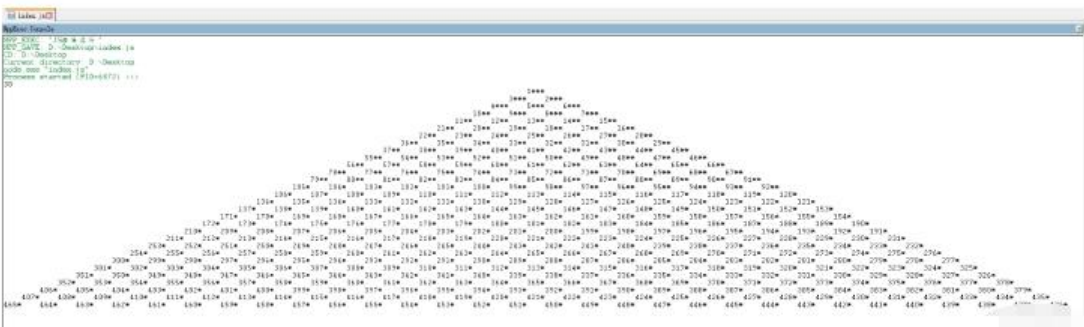
用例

输入	2
输出	1*** 3*** 2***
说明	符号*表示，数字不满4位时的补位，符号X表示数字之间的空格。注意实际编码时不需要打印X，直接打印空格即可。此处为说明题意，故此加上X。

题目解析

这题没啥好说的，按照要求写就行，就是感觉小华有点欠揍。

贴一下N=30的运行效果图



Java算法源码

```
1 import java.util.Scanner;
2 import java.util.StringJoiner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         int n = sc.nextInt();
8         getResult(n);
9     }
10
11     public static void getResult(int n) {
12         // 每行要打印的数，起始为第一行第一个数1
13         int print = 1;
14
15         for (int i = 1; i <= n; i++) { // 多少行
16             int[] printList = new int[i];
17             for (int j = 0; j < i; j++) { // 每行多少个数字
18                 printList[j] = print++;
19             }
20
21             if (i % 2 == 0) reverse(printList); // 如果是偶数行，则逆序
22
23             StringJoiner printListStr = new StringJoiner(" ");
24             for (int ele : printList) {
25                 StringBuilder sb = new StringBuilder(ele + "");
26                 while (sb.length() < 4) sb.append("**"); // 如果数字不足4位则后面用*补足
27                 printListStr.add(sb);
28             }
29
30             StringBuilder res = new StringBuilder(printListStr.toString());
31             for (int k = 0; k < n - i; k++) {
32                 res.insert(0, " "); // 每行的缩进
33             }
34             System.out.println(res);
35         }
36     }
37
38     public static void reverse(int[] printList) {
39         int l = 0;
40         int r = printList.length - 1;
41
42         while (l < r) {
43             int tmp = printList[l];
44             printList[l] = printList[r];
45             printList[r] = tmp;
46             l++;
47             r--;
48         }
49     }
50 }
```

JS算法源码

```
1  /* Javascript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  rl.on("line", (line) => {
10    const n = parseInt(line);
11
12    let print = 1; // 每行要打印的数, 起始为第一行第一个数1
13    for (let i = 1; i <= n; i++) { // 多少行
14      let printList = [];
15      for (let j = 0; j < i; j++) { // 每行多少个数
16        printList.push(print++);
17      }
18      if (i % 2 === 0) { // 如果是偶数行, 则逆序
19        printList.reverse();
20      }
21      printList = printList.map((ele) => { // 如果数字不足4位则后面用*补足
22        let arr = (ele + "").split("");
23
24        while (arr.length < 4) {
25          arr.push("*");
26        }
27
28        return arr.join("");
29      });
30
31      let res = [];
32      res.push(printList.join("  "));
33      for (let m = 0; m < n - i; m++) { // 每行的缩进
34        res.unshift("  ");
35      }
36      console.log(res.join("\n"));
37    }
38  });
```

Python算法源码

```
1  # 输入获取
2  n = int(input())
3
4
5  # 算法入口
6  def getResult():
7      printNum = 1 # 每行要打印的数, 起始为第一行第一个数1
8      for i in range(1, n + 1):
9          printList = []
10         for j in range(i):
11             # 每行多少个数
12             printList.append(printNum)
13             printNum += 1
14
15         if i % 2 == 0:
16             # 如果是偶数行, 则逆序
17             printList.reverse()
18
19         sArr = []
20
21         for ele in printList:
22             tmp = str(ele)
23
24             # 如果数字不足4位则后面补0补足
25             while len(tmp) < 4:
26                 tmp += "0"
27
28             sArr.append(tmp)
29
30         res = " ".join(sArr)
31
32         for k in range(n - i):
33             # 每行的缩进
34             res = "    " + res
35
36         print(res)
37
38
39 # 调用算法
40 getResult()
```