

题目描述：

小华是个很有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; **$1 \leq N \leq 30$**

输入：2

输出描述：

XXXX1***

3***XXXX2***

补充说明：

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

示例 1

输入：

2

输出：

1***

```

3***      2***
num = int(input())
num_temp = 0
row_num = 1
list1 = [i + 1 for i in range((num * num + num) // 2)]
for i in range(num):
    num_list = [list1.pop(0) for i in range(i + 1)]
    if i % 2 == 0:
        resa_t = f'" " * (num - num_temp - 1) * 4'
        for j in num_list:
            string1 = str(j) + (4 - len(str(j))) * '*'
            # resa_t += {num}
            resa_t += f'{string1}' + ' ' * 4
        resa_t = resa_t[:-4]
        resa_t += f'" " * (num - num_temp - 1) * 4'
        print(resa_t)
    else:
        num_list = num_list[::-1]
        resa_t = f'" " * (num - num_temp - 1) * 4'
        for j in num_list:
            string1 = str(j) + (4 - len(str(j))) * '*'
            # resa_t += {num}
            resa_t += f'{string1}' + ' ' * 4
        resa_t = resa_t[:-4]
        resa_t += f'" " * (num - num_temp - 1) * 4'
        print(resa_t)
    num_temp += 1

```