1、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<=N<=30

输入：2

**输出描述:**

XXXX1\*\*\*

3\*\*\*XXXX2\*\*\*

**示例1**

输入

2

输出

    1\*\*\*

3\*\*\*    2\*\*\*

**备注:**

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

2、现有一字符串仅由 '('，')'，'{'，'}'，'['，']'六种括号组成。

若字符串满足以下条件之一，则为无效字符串：

  ①任一类型的左右括号数量不相等；

  ②存在未按正确顺序（先左后右）闭合的括号。

输出括号的最大嵌套深度，若字符串无效则输出0。  
0≤字符串长度≤100000

**输入描述:**

一个只包括 '('，')'，'{'，'}'，'['，']'的字符串

**输出描述:**

一个整数，最大的括号深度

**示例1**

输入

[]

输出

1

说明

有效字符串，最大嵌套深度为1

**示例2**

输入

([]{()})

输出

3

说明

有效字符串，最大嵌套深度为3

**示例3**

输入

(]

输出

0

说明

无效字符串，有两种类型的左右括号数量不相等

**示例4**

输入

([)]

输出

0

说明

无效字符串，存在未按正确顺序闭合的括号

**示例5**

输入

)(

输出

0

说明

无效字符串，存在未按正确顺序闭合的括号

3、小明负责公司年会，想出一个趣味游戏：  
屏幕给出1～9中任意4个不重复的数字，大家以最快时间给出这几个数字可拼成的数字从小到大排列位于第N位置的数字，其中N为给出的数字中最大的（如果不到这么多个数字则给出最后一个即可）。  
注意：  
1）2可以当做5来使用，5也可以当做2来使用进行数字拼接，且屏幕不能同时给出2和5；  
2）6可以当做9来使用，9也可以当做6来使用进行数字拼接，且屏幕不能同时给出6和9。  
  
如给出：1，4，8，7则可以拼成的数字为：  
1，4，7，8，14，17，18，41，47，48，71，74，78，81，84，87，147，148，178...（省略后面的数字）  
那么第N（即8）个的数字为41。

**输入描述:**

输入为以逗号分隔的描述4个int类型整数的字符串。

**输出描述:**

输出为这几个数字可拼成的数字从小到大排列位于第N（N为输入数字中最大的数字）位置的数字，如果输入的数字不在范围内或者有重复，则输出-1。

**示例1**

输入

1,4,8,7

输出

41

说明

可以构成的数字按从小到大排序为1，4，7，8，14，17，18，41，47，48，71，74，78，81，84，87，147，148，178...（省略后面的数字），故第8个为41

**示例2**

输入

2,5,1

输出

-1

说明

2和5不能同时出现

**示例3**

输入

3,0,9

输出

-1

说明

0不在1到9的范围内

**示例4**

输入

3,9,7,8

输出

39

说明

注意9可以当6使用，所以可以构成的数字按从小到大排序为3，6，7，8，9，36，37，38，39，63，67，68，73，76，78，79，83...（省略后面的数字），故第9个为39