

题目描述：

给定一个正整数 n ，如果能够分解为 $m(m > 1)$ 个连续正整数之和，请输出所有分解中， m 最小的分解。

如果给定整数无法分解为连续正整数，则输出字符串 "N"。

输入描述：

输入数据为一整数，范围为 $(1, 2^{30}]$

输出描述：

比如输入为：

21

输出：

21=10+11

补充说明：

21 可以分解的连续正整数组合的形式有多种

21=1+2+3+4+5+6

21=6+7+8

21=10+11

输出，21=10+11，是最短的分解序列。

示例 1

输入：

21

输出：

21=10+11

说明：

21 可以分解的连续正整数组合的形式有多种

21=1+2+3+4+5+6

$21=6+7+8$

$21=10+11$

因 $21=10+11$ ，是最短的分解序列。所以答案是 $21=10+11$

import sys

```
def find_continues_sum(n):
    for m in range(2,int((2*n)**0.5+1)):
        if (2*n)%m == 0:
            k = (2*n)//m
            if (k-m+1)%2 != 0:
                continue
            a = (k-m+1)//2
            b = (k+m-1)//2
            return "{}={}".format(n,"+".join(str(i) for i in range(a,b+1)))
    return "N"
```

```
num = int(input())
print(find_continues_sum(num))
```