

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 $21=10+11$

```
sum = int(input())
```

```
flag = 0
```

```
h = 2
```

```
while h * (h - 1) < 2 * sum:
```

```
    if 2 * sum % h == 0 and (2 * sum // h - h + 1) % 2 == 0:
```

```
        a = (2 * sum // h - h + 1) // 2
```

```
        flag = 1
```

```
        print(f"{sum}=", end="")
```

```
        for i in range(1, h):
```

```
            print(str(a)+"", end="")
```

```
            a += 1
```

```
        print(a)
```

```
        break
```

```
    h += 1
```

```
if flag == 0:
```

```
    print("N")
```