A. Help Farmer

Once upon a time in the Kingdom of Far Far Away lived Sam the Farmer. Sam had a cow named Dawn and he was deeply attached to her. Sam would spend the whole summer stocking hay to feed Dawn in winter. Sam scythed hay and put it into haystack. As Sam was a bright farmer, he tried to make the process of storing hay simpler and more convenient to use. He collected the hay into cubical hay blocks of the same size. Then he stored the blocks in his barn. After a summer spent in hard toil Sam stored $A \cdot B \cdot C$ hay blocks and stored them in a barn as a rectangular parallelepiped A layers high. Each layer had B rows and each row had C blocks.

At the end of the autumn Sam came into the barn to admire one more time the hay he'd been stacking during this hard summer. Unfortunately, Sam was horrified to see that the hay blocks had been carelessly scattered around the barn. The place was a complete mess. As it turned out, thieves had sneaked into the barn. They completely dissembled and took away a layer of blocks from the parallelepiped's front, back, top and sides. As a result, the barn only had a parallelepiped containing $(A - 1) \times (B - 2) \times (C - 2)$ hay blocks. To hide the evidence of the crime, the thieves had dissembled the parallelepiped into single $1 \times 1 \times 1$ blocks and scattered them around the barn. After the theft Sam counted n hay blocks in the barn but he forgot numbers A, B и C.

Given number n, find the minimally possible and maximally possible number of stolen hay blocks.

Input

The only line contains integer n from the problem's statement ($1 \leq n \leq 10^9$).

Output

Print space-separated minimum and maximum number of hay blocks that could have been stolen by the thieves.

Note that the answer to the problem can be large enough, so you must use the 64-bit integer type for calculations. Please, do not use the %lld specificator to read or write 64-bit integers in C++. It is preferred to use cin, cout streams or the %I64d specificator.

Examples

input
```
4
```
output
```
28 41
```

input
```
7
```
output
```
47 65
```

input
```
12
```
output
```
48 105
```

Note

Let's consider the first sample test. If initially Sam has a parallelepiped consisting of $32 = 2 \times 4 \times 4$ hay blocks in his barn, then after the theft the barn has $4 = (2 - 1) \times (4 - 2) \times (4 - 2)$ hay blocks left. Thus, the thieves could have stolen $32 - 4 = 28$ hay blocks. If Sam initially had a parallelepiped consisting of $45 = 5 \times 3 \times 3$ hay blocks in his barn, then after the theft the barn has $4 = (5 - 1) \times (3 - 2) \times (3 - 2)$ hay blocks left. Thus, the thieves could have stolen $45 - 4 = 41$ hay blocks. No other variants of the blocks' initial arrangement (that leave Sam with exactly 4 blocks after the theft) can permit the thieves to steal less than 28 or more than 41 blocks.