Over Optimization

Time Limit: 2.0s **Memory Limit:** 64M

There exists so many problems on MCPT that require a lot of optimization and implementing. To solve those problems, you must use algorithms that are more efficient like the ones we learned last week, but those will be required in the later problems. However, let's assume that we are in a theoretical world where optimizing code is a lot simpler. In this world, we define `optimizing` as taking a integer time, square rooting it and rounding down, then subtracting by 5, which will give you the optimized time. However, if your number ends up being less than or equal to 0, that means that your code cannot be optimized.

Given the original time that some code takes to run, print out the amount of time it will take after optimizing or `impossible` if the time cannot be optimized.

Input Specification

One integer N $(1 \leq N \leq 10^5)$, the original time that the code takes to run.

Output Specification

One integer representing the time the code takes after it is optimized, or `impossible` if it cannot be optimized.

Sample Input 1

```
49
```

Sample Output 1

```
2
```

Sample Input 2

```
24
```

Sample Output 2

```
impossible
```