A Another Rock-Paper-Scissors Problem

Sonny uses a very peculiar pattern when it comes to playing rock-paper-scissors. He likes to vary his moves so that his opponent can't beat him with his own strategy.

Sonny will play rock (R) on his first game, followed by paper (P) and scissors (S) for his second and third games, respectively. But what if someone else is using the same strategy? To thwart those opponents, he'll then play paper to beat rock, scissors to beat paper, and rock to beat scissors, in that order, for his 4th through 6th games. After that, he'll play scissors, rock, and paper for games 7–9 to beat anyone copying his last set of moves. Now we're back to the original order—rock, paper, scissors—but instead of being predictable and using the same moves, do you know what would be better? You guessed it! Sonny then plays the sequence of moves that would beat anyone trying to copy his whole strategy from his first move, and on it goes...

To recap, in symbolic form, Sonny's rock-paper-scissors moves look like this:

```
R P S PSR SRP PSRSRPRPS SRPRPSPSR PSRSRPRPSSRPRPSPSRRPSPSRSRP ...
```

The spaces are present only to help show Sonny's playing pattern and do not alter what move he'll play on a certain game.

Naturally, your job is to beat Sonny at his own game! If you know the number of the game that you'll be playing against Sonny, can you figure out what move you would need to play in order to beat him?

A.1 Input

Each line of the input contains a single integer N, $1 \leq N \leq 10^{12}$, the number of the game you'll be playing against Sonny. An integer $N = 1$ indicates it would be Sonny's first game, $N = 7$ indicates it would be the 7th game, and so forth. The input terminates with a line with $N = 0$. For example:

```
1
7
33
0
```

Warning: N may be large enough to overflow a 32-bit integer, so be sure to use a larger data type (i.e. `long` in Java or `long long` in C/C++) in your program.

A.2 Output

For each test case, output a single line which contains the letter corresponding to the move you would need to play to beat Sonny on that game. For example, the correct output for the sample input above would be:

```
P
R
S
```