

# Challenge 1

*Time Taken: Under 30 Minutes*

- This one took the longest time to do, and I spent some time trying multiple ways to solve it, including first dividing numbers into 5 different kinds: Digits, Twenties, Digits at tens place, Larger numbers (hundreds, thousands, millions) but that remained unstable and I had to change the approach.
- Passes all given test cases and some additional ones as well, but fails complicated numbers like: `"one million four hundred fifty six thousand seven hundred eighty nine"`. Investing a little more time into it would probably solve that issue.

# Challenge 2

*Time Taken: Under 20 Minutes*

- Started with a simple brute force solution that reverses the whole string, then finds the longest common substring between the original and reversed string. This was a bit slow and I had to change the approach by employing memoization to store the results of the longest common substring between the original and reversed string, which made it significantly faster as it reduced the number of duplicate calculations.
- Passes all given test cases and some additional ones as well.

# Challenge 3

*Time Taken: Under 15 Minutes*

- This was the easiest of the three challenges, and I was able to solve it in a single go. Started with collecting all the pieces with the same length together, then matching them only with the ones that add up to 6. This was a simple and effective approach.
- After finding a working solution in <5 minutes, I realized that I should also compare what word the current piece matches to in the dictionary, that involved creating another method that checked the piece against the dictionary to find the word it matches to.
- Passes all given test cases and some additional ones as well.