This program is to decode words by pattern matching.

Every word has a pattern.

For example," bullet" has a pattern "012234", that means the third letter and fourth are the same and all the other letters are different.

Since the pattern does not change when encrypting, the encrypted word shares the same pattern as original word.

Therefore, I just compare the encrypted word with the dictionary. For example, encrypted word "bckkge" has a pattern 012234, then we check the words in dictionary with the same pattern. "bullet", "fatter" are the potential words.

Then encrypted letter b could have potential letter" b" and "f"

Next time, if we find another word with letter b, and b have potential letter "f" and "g", then encrypted "b" would probably be "f" in the original text.

However, we cannot just simply combine two words to get "f". And also, in some cases, the word in original text would not be in dictionary. For example, my name "Lu" is not a valid word. That interrupts the results. Therefore, I choose to count the total times appear among all words.

For example, the final count would be like

A:{M:9 times, N: 10 times} ---------------------------choose N for A

B:{B:3 times, F:8 times, G:5 times }------------------choose F for B

And every letter has the most potential letter. It is still possible for duplicate letter correspond to single letter. Then I use guess to solve that.

That is how it works.

Thanks for reading

Idea from

Resources: https://inventwithpython.com/cracking/chapter17.html