A guilty key

이문제는 암호를 decode, encode하는거가 기본원리.

**>>>** cipher = Cipher('ABCD', '1AX3S1M2PYZ')

**>>>** cipher.grid

[['-', 'A', 'X', '-'], ['-', '-', 'S', '-'], ['M', '-', '-', 'P'], ['Y', 'Z', '-', '-']]

Class cipher:

Cipher.grid는 cipher(‘글자’, ‘암호’)

암호는 1AX3S1M2PYZ 이형태처럼 숫자와 알파벳이 같이있고, 숫자는 ‘-‘이거로 표현

1AX3S1M2PYZ => (‘-‘ \* 1) + A + X + (‘-‘ \* 3) + S + (‘-‘ \* 1) + M + (‘-‘ \* 2) + P + Y + Z

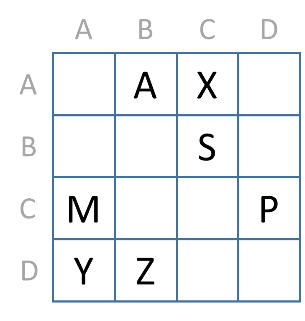
그리고 글자의 숫자만큼 나눈다

[['-', 'A', 'X', '-'], ['-', '-', 'S', '-'], ['M', '-', '-', 'P'], ['Y', 'Z', '-', '-']]

Cipher.map

**>>>** cipher.map

{'A': 'AB', 'X': 'AC', 'S': 'BC', 'M': 'CA', 'P': 'CD', 'Y': 'DA', 'Z': 'DB'}

이거는 위에 그림

이거인데 cipher.grid의 결과를

[['-', 'A', 'X', '-'],

['-', '-', 'S', '-'],

['M', '-', '-', 'P'],

['Y', 'Z', '-', '-']] 🡺 x, y축 함수형태로 표현

이렇게 배치를 해 놓은 것.

그리고 결과값을 좌표값으로, 예를 들어서 A는 세로줄A, 가로줄B 에 위치하여서 {A : AB}

**>>>** cipher.encode('spam')

'BCCDABCA'

**>>>** cipher.decode('BCCDABCA')

'SPAM'

Cipher.encode는 위에 cipher.map의 결과값을 이용해서 encode, decode

# Assertion Error

**>>>** cipher.encode('eggs')

Traceback (most recent call last):

AssertionError: invalid message

**>>>** cipher.decode('BCCDBACA')

Traceback (most recent call last):

AssertionError: invalid message

1. 만약 단어에서 해당하는 알파벳이 없는경우
2. 해당좌표에 알파벳이없는경우
3. 아마 한 개더있는거같은데 decode안에있는 암호의 글자수가 홀수일때도 error가 뜰거같아요