The problem is a modified and simplified implementation of a ‘Morbit’ cipher (<https://sites.google.com/site/cryptocrackprogram/user-guide/cipher-types/other/morbit>)

Problem Flow:

Take input ->

convert to morse as per provided map ->

replace spaces with filler char (here, ‘x’) and also append it at end of each char ->

if len of morse text is odd, drop last char (which will always be the filler char) ->

make pairs of 2 chars and compare with key\_pairs array, containing every possible combination of the used chars ->

convert to decimal bits as per key ((index + 1) for resp. key pair array) ->

output string of numeric.

For simplicity, here key is ‘123456789’. char Key[i] means key\_pairs[ int(key[i]) – 1]

**NOTE:** Adhering to format for input is important. Input must contain only alphabetic characters with an exception of whitespace char.

Example: (for key ‘123456789’ pointing to key\_pairs = ['..', '.-', '.x', '-.', '--', '-x', 'x.', 'x-', 'xx'])

1 2 3 4 5 6 7 8 9

Input: ROUNDTABLE

In morse: .-.x---x..-x-.x-..x-x.-x-...x.-..x.x

In morbit: 235616481876417433

**Additonal:** instead of a set key, we can make the participant input it, which must be a permutation of 1 to 9 (9 pairs) in total and map them accordingly.