Side note:

I’m gonna interchange encoding for language sometimes.

Meownese is a language that I made up mainly so that I could secretly and cutely communicate with some of my friends on a server (It lasted for a day before we were told to move our shenanigans into the spam channel). It took me about two class periods to come up with the rules for this language and how it was going to work. The goals that I had for this encryption is as follows:

- Make is very hard for a normal person to learn how to decrypt the encryption
- Make the language seem like it is predictable, while also not being so

In order to encrypt text to meownese, it must first be converted to morse code, then to alternating code – An intermediate step I made up to make the conversion a bit easier –, then finally to meownese.

If I wanted to encrypt the phrase “timmy cat”, the first step would be to convert it to morse code (International Morse Code). The resulting code would be -*..*--*--*-.--_-.-.*.-*- (‘ * ’ denoting spaces between letters, and ‘ _ ‘ denoting spaces between words).

Next is to turn it into alternating code. Each message in alternating code has two 2 dimensional lists:

A list containing the number of morse code signals in each letter (in code, this list is represented as 1 dimensional).

<table>
<tr><td>Word 1</td><td><table><tr><td>-</td><td>..</td><td>--</td><td>--</td><td>-.--</td></tr><tr><td>1</td><td>2</td><td>2</td><td>2</td><td>4</td></tr></table></td></tr>
<tr><td>Word 2</td><td><table><tr><td>-.-.</td><td>.-</td><td>-</td></tr><tr><td>4</td><td>2</td><td>1</td></tr></table></td></tr>
</table>

A 2 dimensional list containing the “signals in a row” in each word. In other words, it contains the durations a signal repeats itself in each word.

<table>
<tr><td>Word 1</td><td><table><tr><td>(no dot)</td><td>-</td><td>..</td><td>-----</td><td>.</td><td>--</td></tr><tr><td>0</td><td>1</td><td>2</td><td>5</td><td>1</td><td>2</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td></tr>
<tr><td>Word 2</td><td><table><tr><td>(no dot)</td><td>-</td><td>.</td><td>-</td><td>..</td><td>--</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>2</td><td>2</td></tr></table></td></tr>
</table>

The resulting alternating code is:

- Word 1
  - Letters:
    - 1 2 2 2 4
  - Signal:
    - 0 1 2 5 1 2
- Word 2
  - Letters:
    - 4 2 1
  - Signal:
    - 0 1 1 1 2 2

Things get more complicated when converting alternating code to meownese
There's a few steps to converting alternating code to meownese:

For each word, starting with 'e's, for each letter add an 'e' for the number of times the letter indicates, then swap between using 'e's and 'o's after each letter. BUT if the starting signal in a word is a 0, then start the alternation with an 'o'.

<table>
<tr><td>Word 1</td><td><table><tr><td>1 (signal starts with 0)</td><td>2</td><td>2</td><td>2</td><td>4</td></tr><tr><td>o</td><td>ee</td><td>oo</td><td>ee</td><td>oooo</td></tr></table></td></tr>
<tr><td>Word 2</td><td><table><tr><td>4 (signal starts with 0)</td><td>2</td><td>1</td></tr><tr><td>oooo</td><td>ee</td><td>o</td></tr></table></td></tr>
</table>

Next, you want to make groups of 'e's and 'o's in each word equal to the duration of the signals in each word. Combine any singular groups with ones directly to the left of it. Then add m to the start and w to the end of each group:

Word 1	0 + 1	2	5 + 1	2
	mow	meew	mooeeoow	moow
Word 2	1 + 1 + 1	2	2	
	mooow	moew	meow	

Second to last, you want to add an ‘r’ before the first signal duration that is 1.

Word 1	1 (1 by itself is an exception)	2	5 r 1	2
	mow	meew	mooeeorow	moow
Word 2	r 1 + 1 + 1	2	2	
	mrooow	moew	meow	

Lastly, we need indicate spaces between words, and to do so, we want to join the first and last meow of each word together:

Mow meew mooeeorow moowrooow moew meow