Worksheet Activity: Text Compression

Name:_____

Directions:
Compress the verses of the tongue twister shown below, using the techniques of text compression that you've learned. Use the Pitter Patter example as your guide.

Notes:
A space (indicated by _) counts as a character. So do punctuation marks. You can ignore carriage returns. The grids provided for the dictionary and compressed text are meant to make counting easier for you.

A_tutor_who_tooted_the_flute_

Tried_to_tutor_two_tooters_to_toot._

Said_the_two_to_their_tutor,_

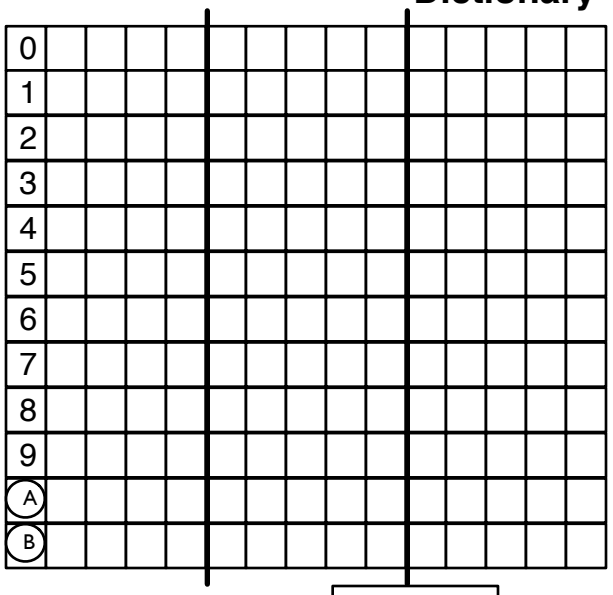
"Is_it_harder_to_toot_

Or_to_tutor_two_tooters_to_toot?"

149 bytes

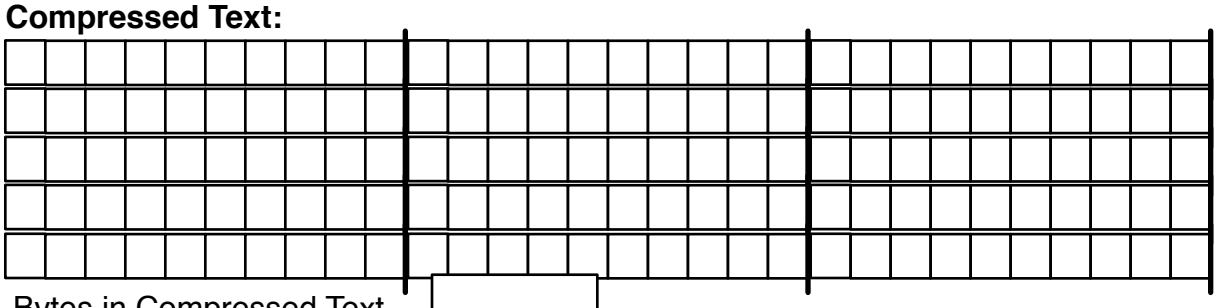
How to count:
If there's a character in a square it counts as a byte. Index numbers on the dictionary also count as bytes. Notice that once you go past the 10th dictionary entry you need to invent a single character symbol as a dictionary index.

Dictionary



Bytes in Dictionary

Compressed Text:



Bytes in Compressed Text

Bytes in Text		Bytes in Dictionary		Total
	+		=	

Percent Compressed
(1-(total/149))*100:

Text Compression :: Pitter Patter example

Here is the "Pitter Patter" poem example we did in class. The original poem, dictionary, and compressed text is shown.

Recall that to arrive at this final dictionary, we had to go through several revisions looking for patterns and realizing new ones.

Also note that the dictionary is self-referencing, with later entries referencing earlier ones.

Original Poem

Pitter_Patter_
Pitter_Patter_
Listen_to_the_rain_
Pitter_Patter_
Pitter_Patter_
On_the_window_pane

121 bytes

Dictionary

0	t	t	e	r	_								
1	n	_	t										
2	h	e	_										
3	p	a											
4	P	i	0	3	0								
5	4	4											
6													
7													

26 bytes

Compressed Text

5	L	i	s	t	e	1	o	_	t	2	r	a	i	n	_	5	O	1	2	w	i	n	d	o	w	_	3	n	e
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

30 bytes (chars including spaces)

Total = 30+26 = 56 bytes = ~54% compression

Note: There is actually one more 2-character pattern that occurs in the text - "in" The pattern "in" occurs twice and it was not included in the dictionary because it would actually increase the total number of bytes in the compressed text rather than reduce it. Do you see why?

How it might actually be stored:

Recall that the ASCII code for english text requires only 7 bits since there are only 128 ASCII characters. But a byte has 8-bits. Therefore each byte carries with it one unused bit. We can use that last bit to indicate whether the byte should be read as a normal ASCII character or as a dictionary entry. Additionally, the dictionary can use the 8th bit to denote when the next dictionary entry is starting.

Is this type of compression actually used?

Yes. It's called Lempel-Ziv compression or "LZ" compression for short. Lempel and Ziv are Isreali computer scientists who invented it in the late 70s. An MIT proffessor, Terry Welch, improved LZ compression slightly and so the current version in use is called LZW comprression. It is the compression scheme used in ".zip" files. Of course the real Limpel-Zev compression works by looking for patterns in the 0s and 1s since a computer program that might do the compression would be ignorant of language. LZ compression works very well for any kind of file in which there are a lot of patterns. Thus, it works very well for compressing plain text files since languages tend to have a lot of patterns.