Activity – Exploring Compression of Text

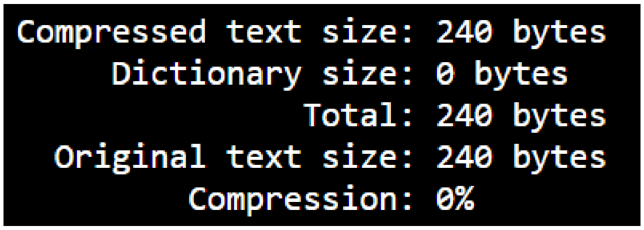
Getting Started:

- Bring up a web-browser and navigate to
  - https://studio.code.org/s/text-compression/lessons/1/levels/2
  - (an active link to this is listed on the class website)
- If you haven't done so previously, watch the video associated with this app.

By default, the text in the simulator should start as some of the lyrics to the Aloe Blacc song "Wake Me Up" If it isn't there select it from the "Choose text" drop down menu.

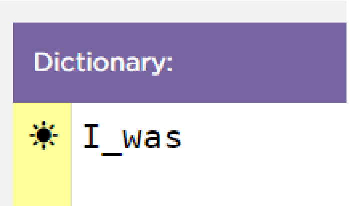
1. Read the lyrics in the "Compressed:" panel on the left edge.

2. Notice that the current size of this file is 240 bytes. This is because there are 240 characters in the lyrics. If each is recorded as its one-byte ascii value, this requires 240 bytes total.



```
Compressed text size: 240 bytes
     Dictionary size: 0 bytes
               Total: 240 bytes
  Original text size: 240 bytes
         Compression: 0%
```

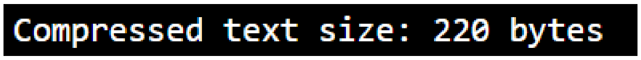
3. In the dictionary panel add "I_was" next to the sun icon.



Dictionary:

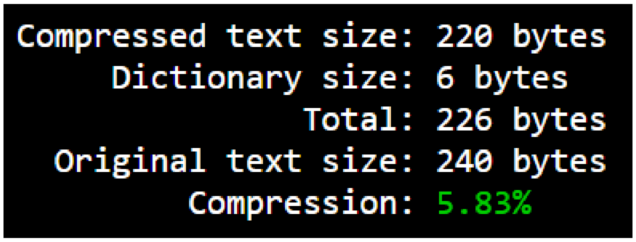
☀ I_was

4. Notice that each of the five occurrences of the five-character phrase "I_was" has been replaced by the sun character. This means that 5x5=25 bytes went away (each of the five-character phrases is gone) and was replaced by a one character/one byte sun. This means that we now have a text size of 220 because 240 – 25 bytes + bytes = 220.



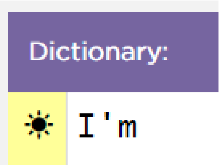
```
Compressed text size: 220 bytes
```

5. But we also have to record our “translation.” That is, we have to tell the computer that we replaced “I_was” with the sun. This takes 5+1 = 6 bytes of information. So while we eliminated 20 bytes from the main text, we do still have to store six bytes to tell us how to get it back. In the end, the total size of our compressed file is 226 byes for a savings of 5.83%



```
Compressed text size: 220 bytes
     Dictionary size: 6 bytes
               Total: 226 bytes
  Original text size: 240 bytes
         Compression: 5.83%
```

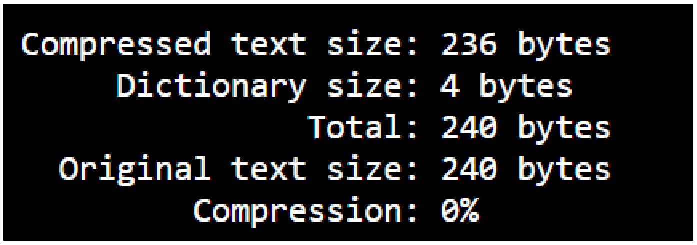
6. Not all replacements save us anything. Change the dictionary so that it uses the sun to replace the phrase “I’m”:



Dictionary:

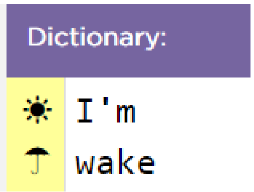
☀ I'm

7. In this situation we got rid of six characters (two occurrences of the three-character phrase I’m) and replaced them with two characters. We reduced the size of our text by a total of four characters. However, we had to add this to the dictionary which requires three characters for the phrase plus one character for the umbrella. This requires the addition of four characters to the dictionary. The net savings is zero:



```
Compressed text size: 236 bytes
     Dictionary size: 4 bytes
               Total: 240 bytes
  Original text size: 240 bytes
         Compression: 0%
```

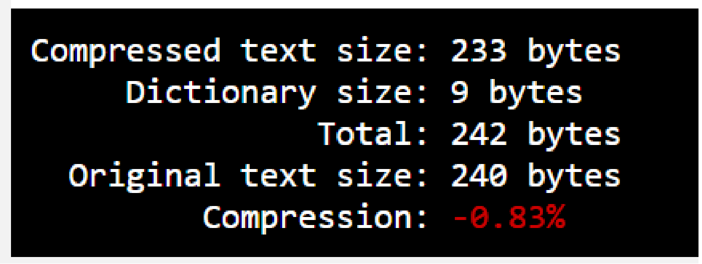
8. In fact, some replacements can even INCREASE the size of the file. Add the word “wake” to the dictionary.



Dictionary:

☀ I'm
☂ wake

9. In this case we had a net change of three characters in the text (drop one four character word but add one one character symbol) while having a net change of five characters in the dictionary (add the four character word and it's one character replacement). We actually increased the total file by two characters.



```
Compressed text size: 233 bytes
     Dictionary size: 9 bytes
               Total: 242 bytes
  Original text size: 240 bytes
         Compression: -0.83%
```

10. Play around with adding different phrases to the dictionary.

- A score between 25 and 30% is pretty good.
- A score between 30% and 35% is excellent.
- My top score was 35.83. I have put an answer key on the website for you to consult if you want to see what I did.

11. Change the original text to the lyrics for "I Need a Dollar"

- A score between 30 and 35 is good.
- A score between 35 and 40 is better.
- A score between 40 and 45 is excellent.
- My top score was 45.19.

12. Play with other text and see what you can learn/figure out.