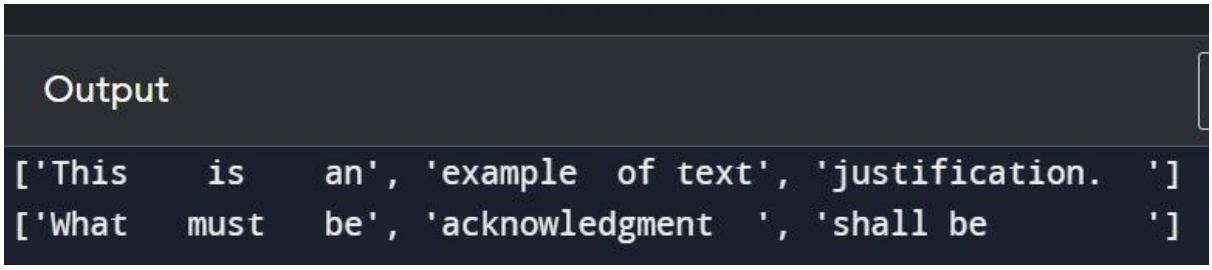
181. Given an array of strings words and a width maxWidth, format the text such that each line has exactly maxWidth characters and is fully (left and right) justified. You should pack your words in a greedy approach; that is, pack as many words as you can in each line. Pad extra spaces ' ' when necessary so that each line has exactly maxWidth characters. Extra spaces between words should be distributed as evenly as possible. If the number of spaces on a line does not divide evenly between words, the empty slots on the left will be assigned more spaces than the slots on the right. For the last line of text, it should be left-justified, and no extra space is inserted between words. A word is defined as a character sequence consisting of non-space characters only. Each word's length is guaranteed to be greater than 0 and not exceed maxWidth. The input array words contains at least one word.

Program:

```
def full_justify(words, maxWidth):
    result = []
    line = []
    total_length = 0

    for word in words:
        if total_length + len(word) + len(line) > maxWidth:
            for i in range(maxWidth - total_length):
                line[i % (len(line) - 1 or 1)] += ' '
            result.append(''.join(line))
            line = []
            total_length = 0

        line.append(word)
        total_length += len(word)

    result.append(' '.join(line).ljust(maxWidth))

    return result

# Example 1
words1 = ["This", "is", "an", "example", "of", "text", "justification."]
maxWidth1 = 16
output1 = full_justify(words1, maxWidth1)
```

```
print(output1)


# Example 2
words2 = ["What", "must", "be", "acknowledgment", "shall", "be"]
maxWidth2 = 16
output2 = full_justify(words2, maxWidth2)
print(output2)
```

Output:



Timecomplexity: O(n * L^2)