I have provided sufficient explanation in code. I will refrain from elaborating on the screen messages…

TEST SCENARIO 1

```
//THESE ARE ALL TEST CASES.....SELECT ONE ONLY!
string text = "The quick brown fox jumps over the lazy dog";
//String text="";
//String text = " This is a test but making it a bit longer!";
//String text = "My";
//String text = "My name is Amit Amlani. This is a sligthly longer test to see if the text can be spanr
//String text = "This will be 16.";
//String text = "Thiswillbetesting  16 testing."
```

*****OUTPUT*****

```
Welcome to Online IDE!! Happy Coding :)
Your line getting bigger:The(3 chars inc white space)
Words after truncation: 1
Your line getting bigger:The·quick(9 chars inc white space)
Words after truncation: 2
Your line getting bigger:The·quick·brown(15 chars inc white space)
Words after truncation: 3
Your line getting bigger:The·quick·brown·fox(19 chars inc white space)
Words after truncation: 4
Rolled back to(due to exceeding 16):The·quick·brown
Length rolled back line: 15
This word getting truncated:fox
The current buffer is: 1
Number words to accomodate for: 3(The·quick·brown)
Last word in tokenizer should be same as truncated word: fox
What is wordcount here: 4
What is truncated word count: 3(The·quick·brown)  15 chars
The current buffer is: 1
The·quick·brown   =>qualifies for 1 padding at front since it has:3 words
CURRENT LENGTH of line: 15
NEW LENGTH of line after formatting: 16
Completed line:·The·quick·brown
This word will be carried over to next line:fox

Your line getting bigger:fox·jumps(9 chars inc white space)
Words after truncation: 2
Your line getting bigger:fox·jumps·over(14 chars inc white space)
Words after truncation: 3
Your line getting bigger:fox·jumps·over·the(18 chars inc white space)
Words after truncation: 4
Rolled back to(due to exceeding 16):fox·jumps·over
Length rolled back line: 14
This word getting truncated:the The
current buffer is: 2
Number words to accomodate for: 3(fox·jumps·over)
Last word in tokenizer should be same as truncated word: the
What is wordcount here: 7
What is truncated word count: 3(fox·jumps·over)  14 chars The
current buffer is: 2
CURRENT LENGTH of line: 14
```

Qualified for extra padding
1 extra padding between:fox·jumps·over
It will now process truncated string with extra 1 padding between the words:fox·jumps·over
Your line getting bigger:fox(3 chars inc white space)
Your line getting bigger:fox··jumps(10 chars inc white space)
Your line getting bigger:fox··jumps··over(16 chars inc white space)
NEW LENGTH of line after formatting: 16
Completed line:fox··jumps··over
This word will be carried over to next line:the

Your line getting bigger:the·lazy(8 chars inc white space)
Words after truncation: 2
Your line getting bigger:the·lazy·dog(12 chars inc white space)
Words after truncation: 3
Finally completed the last line:the·lazy·dog
Left over StringJoiner: 3 word(s)=> the·lazy·dog (12 chars inc white space)
Total running words: 9
Buffer is: 4
It will now process the string with extra: 2 padding between words:the·lazy·dog
your line getting bigger:the(3 chars inc white space will NOT exceed 16) your line
getting bigger:the···lazy(10 chars inc white space will NOT exceed 16) your line
getting bigger:the···lazy···dog(16 chars inc white space will NOT exceed 16)
completed line:the···lazy···dog
NEW LENGTH of line:16

****THIS WILL PRINT ENTIRE TEXT**********

•The·quick·brown fox••jumps••over

the•••lazy•••dog

****

** Process exited - Return Code: 0 **

********EXPECTED OUTCOME********

```
["the  quick brown", # 1 extra space on the left
"fox  jumps  over", # 2 extra spaces distributed evenly
"the   lazy   dog"] # 4 extra spaces distributed evenly
```

N.B There is considered to be an error in the question given…
Placing 4 extra spaces between the words will give 14 characters including white spaces….

the••lazy••dog

TEST SCENARIO 2 AN EMPTY STRING

```
//THESE ARE ALL TEST CASES......SELECT ONE ONLY!
//String text = "The quick brown fox jumps over the lazy dog";
String text="";
//String text = " This is a test but making it a bit longer!";
//String text = "My";
//String text = "My name is Amit Amlani. This is a sligthly longer test to see if the text can be spanr
//String text = "This will be 16.";
//String text = "Thiswillbetesting  16 testing."
```

*****OUTPUT*****

Welcome to Online IDE!! Happy Coding :) Finally
completed the last line:
Left over StringJoiner: 0 word(s)=> (0 chars inc white space)
Total running words: 0 Buffer is: 16 completed line: NEW
LENGTH of line:0

****THIS WILL PRINT ENTIRE TEXT**********

****

TEST SCENARIO 3 – A STRING WITH PADDING ALREADY AT FRONT. UP TO THE COMMA IT IS EXACTLY LENGTH OF K=16. IT SHOULD FIT EXACT UP TO HERE…

```
//THESE ARE ALL TEST CASES......SELECT ONE ONLY!
//String text = "The quick brown fox jumps over the lazy dog";
//String text="";
String text = "This is a test, but making it a bit longer!";
//String text = "My";
//String text = "My name is Amit Amlani. This is a sligthly longer test to see if the text can be spanr
//String text = "This will be 16.";
//String text = "Thiswillbetesting  16 testing."
```

*****OUTPUT*****

Welcome to Online IDE!! Happy Coding :)
Your line getting bigger:This(4 chars inc white space)
Words after truncation: 1
Your line getting bigger:This·is(7 chars inc white space)
Words after truncation: 2
Your line getting bigger:This·is·a(9 chars inc white space)
Words after truncation: 3
Your line getting bigger:This·is·a·test,(15 chars inc white space)
Words after truncation: 4
Your line getting bigger:This·is·a·test,·but(19 chars inc white space)
Words after truncation: 5
Rolled back to(due to exceeding 16):This·is·a·test,
Length rolled back line: 15

This word getting truncated:but The
current buffer is: 1
Number words to accomodate for: 4(This·is·a·test,)
Last word in tokenizer should be same as truncated word: but
What is wordcount here: 5
What is truncated word count: 4(This·is·a·test,) 15 chars The
current buffer is: 1
This·is·a·test,   =>qualifies for 1 padding at front since it has:4 words
CURRENT LENGTH of line: 15
NEW LENGTH of line after formatting: 16
Completed line:·This·is·a·test,
This word will be carried over to next line:but

Your line getting bigger:but·making(10 chars inc white space)
Words after truncation: 2
Your line getting bigger:but·making·it(13 chars inc white space)
Words after truncation: 3
Your line getting bigger:but·making·it·a(15 chars inc white space)
Words after truncation: 4
Your line getting bigger:but·making·it·a·bit(19 chars inc white space)
Words after truncation: 5
Rolled back to(due to exceeding 16):but·making·it·a
Length rolled back line: 15
This word getting truncated:bit
The current buffer is: 1
Number words to accomodate for: 4(but·making·it·a)
Last word in tokenizer should be same as truncated word: bit
What is wordcount here: 9
What is truncated word count: 4(but·making·it·a) 15 chars The
current buffer is: 1
but·making·it·a   =>qualifies for 1 padding at front since it has:4 words
CURRENT LENGTH of line: 15
NEW LENGTH of line after formatting: 16
Completed line:·but·making·it·a
This word will be carried over to next line:bit

Your line getting bigger:bit·longer!(11 chars inc white space)
Words after truncation: 2
Finally completed the last line:bit·longer!
Left over StringJoiner: 2 word(s)=>   bit·longer! (11 chars inc white space)
Total running words: 10
Buffer is: 5
It will now process the string with extra: 5 padding between words:bit·longer!
your line getting bigger:bit(3 chars inc white space will NOT exceed 16) your line
getting bigger:bit·····longer!(16 chars inc white space will NOT exceed 16)
completed line:bit·····longer!
NEW LENGTH of line:16

****THIS WILL PRINT ENTIRE TEXT**********

.This.is.a.test, it still maintains padding on left of first line as expected.

.but.making.it.a it has now incorporated frontal padding since longer than 1 word…

bit.....longer! 3 characters + 6 padding + 7 characters = 16

****

** Process exited - Return Code: 0 **

TEST SCENARIO 4: 1 WORD

I HAD TO TWEAK THE CODE HERE, BUT IT INVOLVED AN EXTRA IF LOOP, THIS WAS FORTUNATELY REALISED DURING TESTING PHASE….

```
//THESE ARE ALL TEST CASES......SELECT ONE ONLY!
//String text = "The quick brown fox jumps over the lazy dog";
//String text="";
//String text = " This is a test, but making it a bit longer!";
string text = "My";
//String text = "My name is Amit Amlani. This is a sligthly longer test to see if the text can be spanr
//String text = "This will be 16.";
//String text = "Thiswillbetesting  16 testing."
```

*****OUTPUT*****

Welcome to Online IDE!! Happy Coding :)
Your line getting bigger:My(2 chars inc white space)
Words after truncation: 1
Finally completed the last line:My
Left over StringJoiner: 1 word(s)=> My (2 chars inc white space) Total running words: 1
Buffer is: 14
Entering here most likely due to having one or no words in the line So the StringBuilder should be empty:
My =>qualifies for 14 padding at end since it has:1 word(s) completed line:My·····
NEW LENGTH of line:16

****THIS WILL PRINT ENTIRE TEXT**********

My•••••••••••••• 14 padding at end

****

** Process exited - Return Code: 0 **

TEST SCENARIO 5: AN EXTRA LONG SENTENCE, BUT ALSO TO SEE DISTINCTION BETWEEN THE PADDING AND FULL STOP

```
//THESE ARE ALL TEST CASES......SELECT ONE ONLY!
//String text = "The quick brown fox jumps over the lazy dog";
//String text="";
//String text = " This is a test, but making it a bit longer!";
//String text = "My";
String text = "My name is Amit Amlani. This is a sligthly longer test to see if the text can be spanned across multiple lines.";
//String text = "This will be 16.";
//String text = "Thiswillbetesting  16 testing."
```

*****OUTPUT*****

Welcome to Online IDE!! Happy Coding :)
Your line getting bigger:My(2 chars inc white space)
Words after truncation: 1
Your line getting bigger:My·name(7 chars inc white space)
Words after truncation: 2
Your line getting bigger:My·name·is(10 chars inc white space)
Words after truncation: 3
Your line getting bigger:My·name·is·Amit(15 chars inc white space)
Words after truncation: 4
Your line getting bigger:My·name·is·Amit·Amlani.(23 chars inc white space)
Words after truncation: 5
Rolled back to(due to exceeding 16):My·name·is·Amit
Length rolled back line: 15
This word getting truncated:Amlani.
The current buffer is: 1
Number words to accomodate for: 4(My·name·is·Amit)
Last word in tokenizer should be same as truncated word: Amlani.
What is wordcount here: 5
What is truncated word count: 4(My·name·is·Amit) 15 chars The
current buffer is: 1
My·name·is·Amit =>qualifies for 1 padding at front since it has:4 words
CURRENT LENGTH of line: 15
NEW LENGTH of line after formatting: 16
Completed line:·My·name·is·Amit
This word will be carried over to next line:Amlani.

Your line getting bigger:Amlani.·This(12 chars inc white space)
Words after truncation: 2
Your line getting bigger:Amlani.·This·is(15 chars inc white space)
Words after truncation: 3
Your line getting bigger:Amlani.·This·is·a(17 chars inc white space)
Words after truncation: 4

Rolled back to(due to exceeding 16):Amlani.·This·is
Length rolled back line: 15
This word getting truncated:a
The current buffer is: 1
Number words to accomodate for: 3(Amlani.·This·is)
Last word in tokenizer should be same as truncated word: a
What is wordcount here: 8
What is truncated word count: 3(Amlani.·This·is) 15 chars The
current buffer is: 1
Amlani.·This·is =>qualifies for 1 padding at front since it has:3 words
CURRENT LENGTH of line: 15
NEW LENGTH of line after formatting: 16
Completed line:·Amlani.·This·is
This word will be carried over to next line:a

Your line getting bigger:a·sligthly(10 chars inc white space)
Words after truncation: 2
Your line getting bigger:a·sligthly·longer(17 chars inc white space)
Words after truncation: 3
Rolled back to(due to exceeding 16):a·sligthly
Length rolled back line: 10
This word getting truncated:longer The
current buffer is: 6
Number words to accomodate for: 2(a·sligthly)
Last word in tokenizer should be same as truncated word: longer
What is wordcount here: 10
What is truncated word count: 2(a·sligthly) 10 chars
The current buffer is: 6 CURRENT
LENGTH of line: 10
Qualified for extra padding
6 extra padding between:a·sligthly
It will now process truncated string with extra 6 padding between the words:a·sligthly
Your line getting bigger:a(1 chars inc white space)
Your line getting bigger:a·····sligthly(16 chars inc white space)
NEW LENGTH of line after formatting: 16
Completed line:a·····sligthly
This word will be carried over to next line:longer

Your line getting bigger:longer·test(11 chars inc white space)
Words after truncation: 2
Your line getting bigger:longer·test·to(14 chars inc white space)
Words after truncation: 3
Your line getting bigger:longer·test·to·see(18 chars inc white space)
Words after truncation: 4
Rolled back to(due to exceeding 16):longer·test·to
Length rolled back line: 14
This word getting truncated:see
The current buffer is: 2
Number words to accomodate for: 3(longer·test·to)
Last word in tokenizer should be same as truncated word: see
What is wordcount here: 13
What is truncated word count: 3(longer·test·to) 14 chars The
current buffer is: 2
CURRENT LENGTH of line: 14
Qualified for extra padding
1 extra padding between:longer·test·to

It will now process truncated string with extra 1 padding between the words:longer·test·to
Your line getting bigger:longer(6 chars inc white space)
Your line getting bigger:longer··test(12 chars inc white space)
Your line getting bigger:longer··test··to(16 chars inc white space)
NEW LENGTH of line after formatting: 16
Completed line:longer··test··to
This word will be carried over to next line:see

Your line getting bigger:see·if(6 chars inc white space)
Words after truncation: 2
Your line getting bigger:see·if·the(10 chars inc white space)
Words after truncation: 3
Your line getting bigger:see·if·the·text(15 chars inc white space)
Words after truncation: 4
Your line getting bigger:see·if·the·text·can(19 chars inc white space)
Words after truncation: 5
Rolled back to(due to exceeding 16):see·if·the·text
Length rolled back line: 15
This word getting truncated:can
The current buffer is: 1
Number words to accomodate for: 4(see·if·the·text)
Last word in tokenizer should be same as truncated word: can
What is wordcount here: 17
What is truncated word count: 4(see·if·the·text) 15 chars The
current buffer is: 1
see·if·the·text =>qualifies for 1 padding at front since it has:4 words
CURRENT LENGTH of line: 15
NEW LENGTH of line after formatting: 16
Completed line:··see·if·the·text
This word will be carried over to next line:can

Your line getting bigger:can·be(6 chars inc white space)
Words after truncation: 2
Your line getting bigger:can·be·spanned(14 chars inc white space)
Words after truncation: 3
Your line getting bigger:can·be·spanned·across(21 chars inc white space)
Words after truncation: 4
Rolled back to(due to exceeding 16):can·be·spanned
Length rolled back line: 14
This word getting truncated:across The
current buffer is: 2
Number words to accomodate for: 3(can·be·spanned)
Last word in tokenizer should be same as truncated word: across
What is wordcount here: 20
What is truncated word count: 3(can·be·spanned) 14 chars The
current buffer is: 2
CURRENT LENGTH of line: 14
Qualified for extra padding
1 extra padding between:can·be·spanned
It will now process truncated string with extra 1 padding between the words:can·be·spanned
Your line getting bigger:can(3 chars inc white space)
Your line getting bigger:can··be(7 chars inc white space)
Your line getting bigger:can··be··spanned(16 chars inc white space)
NEW LENGTH of line after formatting: 16
Completed line:can··be··spanned
This word will be carried over to next line:across

Your line getting bigger:across·multiple(15 chars inc white space)
Words after truncation: 2
Your line getting bigger:across·multiple·lines.(22 chars inc white space)
Words after truncation: 3
Rolled back to(due to exceeding 16):across·multiple
Length rolled back line: 15
This word getting truncated:lines. The
current buffer is: 1
Number words to accomodate for: 2(across·multiple)
Last word in tokenizer should be same as truncated word: lines.
What is wordcount here: 22
What is truncated word count: 2(across·multiple) 15 chars The
current buffer is: 1
CURRENT LENGTH of line: 15
Qualified for extra padding
1 extra padding between:across·multiple
It will now process truncated string with extra 1 padding between the words:across·multiple
Your line getting bigger:across(6 chars inc white space)
Your line getting bigger:across··multiple(16 chars inc white space)
NEW LENGTH of line after formatting: 16
Completed line:across··multiple
This word will be carried over to next line:lines.

Finally completed the last line:lines.
Left over StringJoiner: 1 word(s)=> lines. (6 chars inc white space)
Total running words: 22
Buffer is: 10
Entering here most likely due to having one or no words in the line
So the StringBuilder should be empty: lines. =>qualifies for 10
padding at end since it has:1 word(s) completed line:lines.·····
NEW LENGTH of line:16

****THIS WILL PRINT ENTIRE TEXT**********

·My·name·is·Amit

·Amlani..This·is height of padding distinguishable to full stop

a·····sligthly longer··test··to

·see·if·the·text can··be··spanned

across··multiple

lines.•••••••••• height of padding distinguishable to full stop

****

** Process exited - Return Code: 0 **

TEST SCENARIO 6: EXACTLY 16 CHARS AND WHITESPACE

```
//THESE ARE ALL TEST CASES.....SELECT ONE ONLY!
//String text = "The quick brown fox jumps over the lazy dog";
//String text="";
//String text = " This is a test, but making it a bit longer!";
//String text = "My";
//String text = "My name is Amit Amlani. This is a sligthly longer test to see if the text can be spanned acr
String text = "This will be 16.";
//String text = "Thiswillbetesting  16 testing."
```

*****OUTPUT*****

Welcome to Online IDE!! Happy Coding :)
Your line getting bigger:This(4 chars inc white space)
Words after truncation: 1
Your line getting bigger:This·will(9 chars inc white space)
Words after truncation: 2
Your line getting bigger:This·will·be(12 chars inc white space)
Words after truncation: 3
Your line getting bigger:This·will·be·16.(16 chars inc white space)
Words after truncation: 4
Finally completed the last line:This·will·be·16.
Left over StringJoiner: 4 word(s)=> This·will·be·16. (16 chars inc white space)
Total running words: 4
Buffer is: 0
It will now process the string with extra: 0 padding between words:This·will·be·16.
your line getting bigger:This(4 chars inc white space will NOT exceed 16) your line getting bigger:This·will(9 chars inc white space will NOT exceed 16) your line getting bigger:This·will·be(12 chars inc white space will NOT exceed 16) your line getting bigger:This·will·be·16.(16 chars inc white space will NOT exceed 16) completed line:This·will·be·16.
NEW LENGTH of line:16

****THIS WILL PRINT ENTIRE TEXT**********

This•will•be•16. Remans exactly same

****

** Process exited - Return Code: 0 **

TEST SCENARIO 7: FIRST LINE EXCEEDS K LIMIT

```
//THESE ARE ALL TEST CASES......SELECT ONE ONLY!
//String text = "The quick brown fox jumps over the lazy dog";
//String text="";
//String text = " This is a test, but making it a bit longer!";
//String text = "My";
//String text = "My name is Amit Amlani. This is a sligthly longer test to see if the text can be spanned acr
//String text = "This will be 16.";
string text = "Thiswillbetesting  16 testing."
```

*****OUTPUT*****

Welcome to Online IDE!! Happy Coding :)

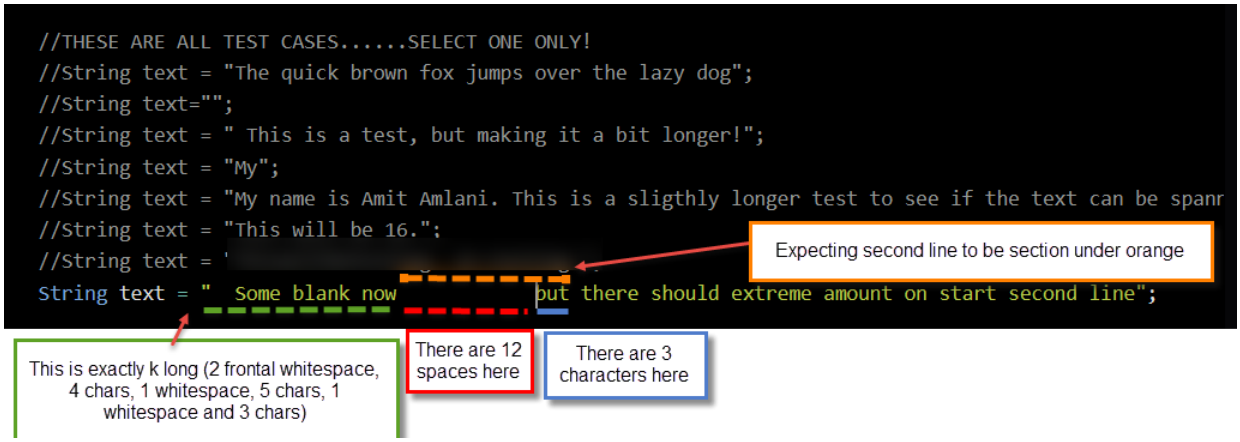
The following word exceeds line limit of k(16): Thiswillbetesting

** Process exited - Return Code: 0 **

*** CODE **

SEE ATTACHMENT

TEST SCENARIO 8: THIS WAS ONLY INTRODUCED SINCE IT WAS REALISED A SCENARIO WAS NOT TESTED AS SUCH… THE FIRST LINE WILL START WITH BLANK SPACES AND THE WORDS WILL END PERFECTLY AT K CONSTRAINT, BUT THE NEXT LINE ALSO HAS NATURAL WHITESPACES IN THE STRING



*****OUTPUT*****

UNFORTUNATELY THIS TIME, IT HAS FAILED SEVERELY!

****THIS WILL PRINT ENTIRE TEXT**********

Some··blank··now it has lost leading padding and hence kept padding inter words

but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second·line a severe mess in line 2 and line 3

****

My initial logic tells me it is something related to having customized paddingTest character… But it is totally unrelated…
It is quite upsetting that this test case was not explored….
This would be great chance to examine the screen outputs. I am hoping its not too detrimental since it is quite upsetting to find this at end of code.

At first instance, the only possible way to sort this is to keeping running total of characters (white space included) of the text (up to point of end of token outputted).
Need to be careful since a token might be repeated multiple times, so how can exact location be found without further errors?
Only possible if the text is stripped off at intervals of the token……
So the initial text has to be stored in a StringBuilder….

ALSO: keep running total of the length of the
tokens…
Anything in between (presumed to be whitespace) has to be placed back in as natural padding…
This has to be done before additional padding since it will compromise k…..

*****OUTPUT*****

Welcome to Online IDE!! Happy Coding :)
Your line getting bigger:Some(4 chars inc white space) it has failed straight away and lost frontal spaces.. This tells me that it is related to the String Tokenizer.. Until now, my assumption was that delimiter is inter-words… So need to find a technique to keep the blank spaces… This would have to be done at start of each line……
Words after truncation: 1
Your line getting bigger:Some·blank(10 chars inc white space)
Words after truncation: 2
Your line getting bigger:Some·blank·now(14 chars inc white space)
Words after truncation: 3
Your line getting bigger:Some·blank·now·but(18 chars inc white space)
Words after truncation: 4
Rolled back to(due to exceeding 16):Some·blank·now
Length rolled back line: 14
This word getting truncated:but
The current buffer is: 2

Number words to accomodate for: 3(Some·blank·now)
Last word in tokenizer should be same as truncated word: but
What is wordcount here: 4
What is truncated word count: 3(Some·blank·now) 14 chars
The current buffer is: 2 CURRENT
LENGTH of line: 14
Qualified for extra padding
1 extra padding between:Some·blank·now
It will now process truncated string with extra 1 padding between the words:Some·blank·now
Your line getting bigger:Some(4 chars inc white space)
Your line getting bigger:Some··blank(11 chars inc white space)
Your line getting bigger:Some··blank··now(16 chars inc white space)
NEW LENGTH of line after formatting: 0
Completed line:Some··blank··now
This word will be carried over to next line:but

Your line getting bigger:but·there(9 chars inc white space)
Words after truncation: 2
Your line getting bigger:but·there·should(16 chars inc white space)
Words after truncation: 3
Your line getting bigger:but·there·should·extreme(24 chars inc white space)
Words after truncation: 4
Rolled back to(due to exceeding 16):but·there·should
Length rolled back line: 16
This word getting truncated:extreme
The current buffer is: 0
Number words to accomodate for: 3(but·there·should)
Last word in tokenizer should be same as truncated word: extreme
What is wordcount here: 7
What is truncated word count: 3(but·there·should) 16 chars The
current buffer is: 0
Your line getting bigger:but·there·should·extreme·extreme·amount(39 chars inc white space)
Words after truncation: 5
Rolled back to(due to exceeding 16):but·there·should·extreme·extreme
Length rolled back line: 32
This word getting truncated:amount
The current buffer is: -16
Number words to accomodate for: 4(but·there·should·extreme·extreme)
Last word in tokenizer should be same as truncated word: amount
What is wordcount here: 8
What is truncated word count: 4(but·there·should·extreme·extreme) 32 chars The
current buffer is: -16
Your line getting bigger:but·there·should·extreme·extreme·amount·amount·on(49 chars inc white space)
Words after truncation: 6
Rolled back to(due to exceeding 16):but·there·should·extreme·extreme·amount·amount
Length rolled back line: 46
This word getting truncated:on
The current buffer is: -30
Number words to accomodate for: 5(but·there·should·extreme·extreme·amount·amount)
Last word in tokenizer should be same as truncated word: on
What is wordcount here: 9
What is truncated word count: 5(but·there·should·extreme·extreme·amount·amount) 46 chars The
current buffer is: -30
Your line getting bigger:but·there·should·extreme·extreme·amount·amount·on·on·start(58 chars inc white
space)
Words after truncation: 7
Rolled back to(due to exceeding 16):but·there·should·extreme·extreme·amount·amount·on·on
Length rolled back line: 52
This word getting truncated:start
The current buffer is: -36
Number words to accomodate for: 6(but·there·should·extreme·extreme·amount·amount·on·on)

Last word in tokenizer should be same as truncated word: start
What is wordcount here: 10
What is truncated word count: 6(but·there·should·extreme·extreme·amount·amount·on·on) 52 chars
The current buffer is: -36
Your line getting bigger:but·there·should·extreme·extreme·amount·amount·on·on·start·start·second(71 chars inc white space) Words after truncation: 8
Rolled back to(due to exceeding 16):but·there·should·extreme·extreme·amount·amount·on·on·start·start
Length rolled back line: 64
This word getting truncated:second
The current buffer is: -48
Number words to accomodate for: 7(but·there·should·extreme·extreme·amount·amount·on·on·start·start)
Last word in tokenizer should be same as truncated word: second
What is wordcount here: 11
What is truncated word count: 7(but·there·should·extreme·extreme·amount·amount·on·on·start·start) 64 chars
The current buffer is: -48 Your line getting
bigger:but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second·line(83 chars inc white space)
Words after truncation: 9
Rolled back to(due to exceeding
16):but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second Length
rolled back line: 78
This word getting truncated:line
The current buffer is: -62 Number
words to accomodate for:
8(but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second)
Last word in tokenizer should be same as truncated word: line
What is wordcount here: 12 What is truncated word count:
8(but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second) 78 chars
The current buffer is: -62 Finally completed the last
line:but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second·line Left
over StringJoiner: 8 word(s)=>
but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second·line (83 chars inc white space)
Total running words: 12
Buffer is: -67
There is one word in last line:line CURRENT LENGTH of line: 4
but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second·line =>qualifies for -67 padding at end since it has:8 word(s) completed
line:but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second·line NEW LENGTH of line:83

****THIS WILL PRINT ENTIRE TEXT********** Some··blank··now
but·there·should·extreme·extreme·amount·amount·on·on·start·start·second·second·line
****

** Process exited - Return Code: 0 **

I have tried extremely hard to remediate this situation… I think it would also begin to add lots of unnecessary code in a way which would not suit the design that I chose to use.
So, the code would be less readable and also perhaps lose its professionalism.

Also, retrospectively, the exercise did promote uniform space between words.. So it would be a contradiction trying to force issue and implement otherwise…

At the moment, if end user has included extra white spaces between words, perhaps they can include a tiny symbol at those locations…