

Test Report (Phase II)

=====

Test	Content	ourResult	correctResult	Status
EmptyFile.txt	null	{}, CharacterCount=0, LineCount=0	{}, CharacterCount=0, LineCount=0	PASSED
SeparatorOnly.txt	Just the space, newline and tab	{}, CharacterCount=0, LineCount=9	{}, CharacterCount=0, LineCount=9	PASSED
SimpleText.txt	SimpleText with simple space only	{the=1, with=1, Hello=1, words.=1, This=2, is=1, simple=1, text=1, just=3, repeated=2}, CharacterCount=66, LineCount=1	{ Hello=1, This=2,is=1, just=3, the=1, simple=1, text=1, with=1,repeated=2, words.=1 },CharacterCount=66, LineCount=1	PASSED
CombinationSeparator.txt	Combination of space,tabs and newlines	{Shrestha=1, in=1, and=1, born=1, Nepal.=1, happy=1, raised=1, I=2, Samita=1, am=2, so=1 }, CharacterCount=48, LineCount= 14	{Shrestha=1, in=1, and=1, born=1, Nepal.=1, happy=1, raised=1, I=2, Samita=1, am=2, so=1 },}, CharacterCount=48, LineCount= 14	PASSED
WordAndNumber.txt	Words and Number	{23=1, a=2, be=1, may=1, I=2, why=3, is=2, do=1, my=1, number=2, 123=1, not=2, and=1, know=2, favorite=1, donot=1, word=1 }, CharacterCount=77, LineCount=1	{23=1, a=2, be=1, may=1, I=2, why=3, is=2, do=1, my=1, number=2, 123=1, not=2, and=1, know=2, favorite=1, donot=1, word=1 },CharacterCount=77, LineCount=1	PASSED
LargerFile.txt	Multiple word, multiple character, Multi Line, Multi Separator and Special characters	{say?=1, train.=1, turned=1, woke=1, Forgot=1, relieved.=1, I=3, Its=1, smile.=1, your=1, out=2, depressed.=1, can=1, late=1, time.=1, okay.=1, of=1, s@id.=1, me=1, Missed=1, up=1, day=1, all=1, But=1, a=1, bo\$\$=1, breath.=1, felt=1, feeling=1, then=1, my=3, lunch.=1, the=1, take=1, what=1, today.=1, And=1, gave=1, work.=1, fine.=1, to=1, Ran=1 }, CharacterCount= 188, LineCount=20	{say?=1, train.=1, turned=1, woke=1, Forgot=1, relieved.=1, I=3, Its=1, smile.=1, your=1, out=2, depressed.=1, can=1, late=1, time.=1, okay.=1, of=1, s@id.=1, me=1, Missed=1, up=1, day=1, all=1, But=1, a=1, bo\$\$=1, breath.=1, felt=1, feeling=1, then=1, my=3, lunch.=1, the=1, take=1, what=1, today.=1, And=1, gave=1, work.=1, fine.=1, to=1, Ran=1 }, CharacterCount= 188, LineCount=20	PASSED
SimpleSeparator	Simple words with one separtor between each string	{will=1, one=1, have=1, words=1, This=1, why=1, is=1, it=1, so=1, separators=1, two=1, between=1 }, CharacterCount=49, LineCount=2	{This=1 ,will=1, have=1, one=1,separators=1, between=1, two=1,words=1, why=1, is=1, it=1, so=1 }, CharacterCount=49, LineCount=2	PASSED

User Story:

The documentation of user story is also posted in: https://github.com/ShresthaSamita/CS5103_qkp232_Samita/issues/1

Title

Count Line Count and Character Count

Description:

I want to extend the previous program and would like to add two other features to count number of lines and number of characters in the given input file.

Acceptance Criteria:

Program is the extension of the Phase I requirement.

For example:

If input file has following line of string:

```
" This is test case to  
test"
```

Expected output should be:

The words are: { This= 1, is= 1, test=2, case=1, to=1 }

The number of lines are : 4

The number of characters are: 20

[NOTE: The output above is unformatted output.]

Test Case based on User Story:

Functional Requirement

From Phase 1: Word Count

- Program shall allow the user to pass the input from the command line for which they want to count the words and its frequency.
- This program is case sensitive; For example, “Test” is not same as “test”.
- The program supports separators like space, tab, newline, and the combination of them.
- The input given should be in textual format.

Additional Requirements: LineCount and CharacterCount

- Program shall count the number of lines and number of characters from the input file.
- For the line count, even if there are no characters but has the separators such as, new line, the line is counted. i.e. every newline character will increment the line count even if there may or may not be any other characters. Check the testcase SeparatorOnly.txt
- The program does not consider syntax or grammatical correctness of the word. For example, S#@tu is still considered a word with 5 number of characters.

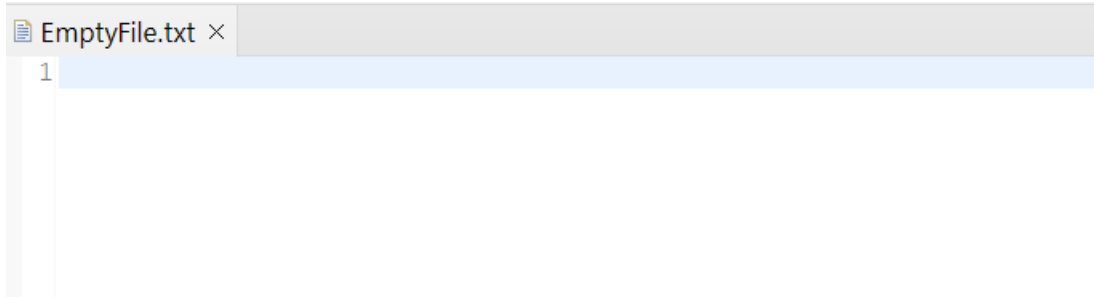
Non-Functional requirement

- Error during the usage, such as file not found, etc. are handled correctly.
- Program is developed using Maven development environment, so, the machine used to run this program must have latest version of MAVEN and JDK installed.
- Works on both MACOS and Windows.

Explanation of Testcase from Test report table above:

1. For Empty Files

[Test for line=0, character=0 and word=0]



Emptyfile.txt is an empty file with no words or separators in it. Hence the test is passed when output is empty. Here, the file does not have any words, characters or separators.

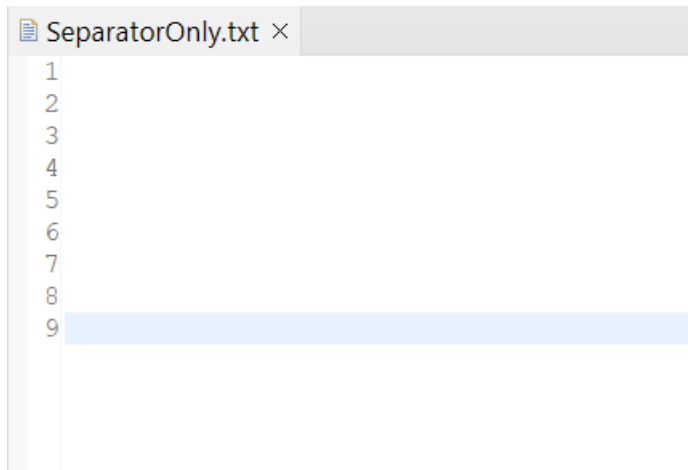
So, it should return an empty HashMap. Similarly, both LineCount and CharacterCount should be 0.

```
@Test
public void testForEmptyFile() throws IOException {

    File file = new File("EmptyFile.txt");
    Map<String, Integer> wordCount = WordCount.getWordCount(new Scanner(file));
    int charCount = CharacterCount.getCharacterCount(new Scanner(file));
    int lineCount = LineCount.getLineCount(new Scanner(file));
    assertTrue((wordCount.isEmpty()) && charCount==0 && lineCount==0 );
}
```

2. For input with separators only (no words or numbers)

[Test for word=0, Character=0 but with newline character]



SeparatorOnly.txt is a file that has number of separators such as space, newline, tab and their combination but does not have any words. This file does not have any words or number but it contains 9 counts of newline. Hence, though HashMap empty for the wordcount and charCount is zero, the number of lines should be returned 9.

```
@Test
public void testForSeparatorsOnly() throws IOException {

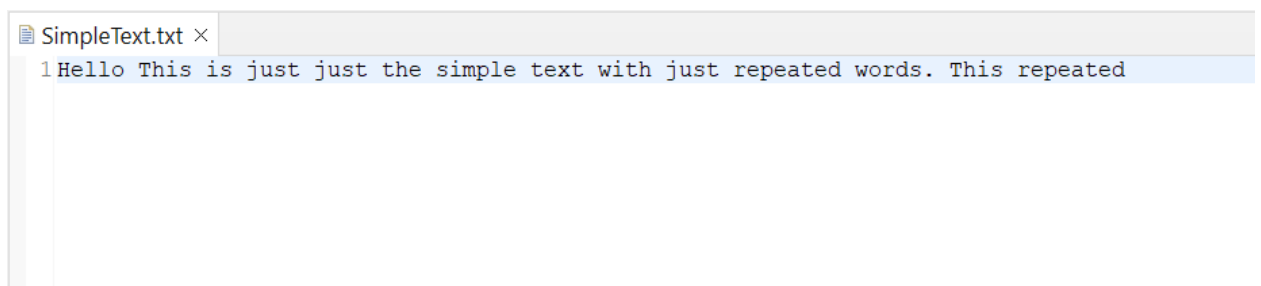
    File file = new File("SeparatorOnly.txt");

    Map<String, Integer> wordCount = WordCount.getWordCount(new Scanner(file));
    int charCount = CharacterCount.getCharacterCount(new Scanner(file));
    int lineCount = LineCount.getLineCount(new Scanner(file));
    assertTrue(wordCount.isEmpty() && charCount==0 && lineCount==9);

}
```

3. For SimpleText with simple space only

[Test for Single Line, Single Separator, Multi word and Multi Character]

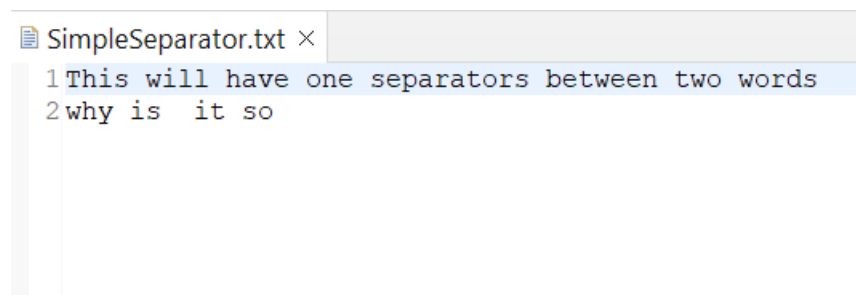


SimpleText.txt contains the simple text with single space only. It is small file with no numbers or multiple separators. This file has some repeated words. For example , This, just, repeated and others are singly present. From the snip above, we can see, it has two lines and 66 characters.

```
assertEquals(correctResult, wordCount);  
assertTrue(charCount==66 && lineCount==1);
```

4. For the input with single space, tab, and newline between words

[Test for Multi Line, Multicharacter, Multiword but single separator]



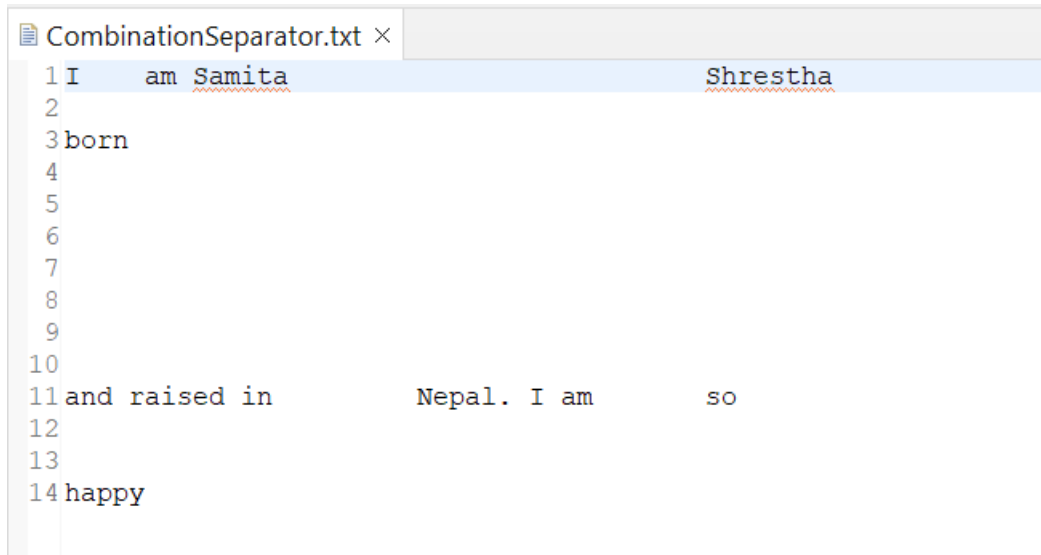
```
SimpleSeparator.txt ×  
1 This will have one separators between two words  
2 why is it so
```

SimpleSeparator.txt is the file that has one separator in between the words but does not have numbers or multiple separators combined. This example is to test the cases where there are multiple lines.

```
assertEquals(correctResult, wordCount);  
assertTrue(charCount==49 && lineCount==2);
```

5. For the input with Combination of space, tabs, and newlines

[Test for Multi Line, Multicharacter, Multiword and combination separator]



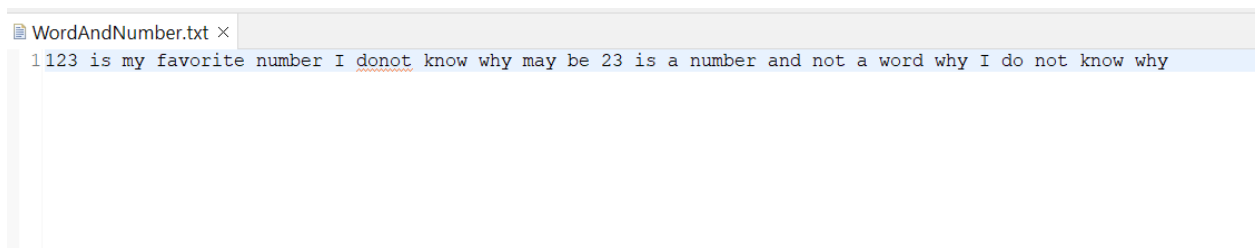
```
CombinationSeparator.txt ×
1 I am Samita Shrestha
2
3 born
4
5
6
7
8
9
10
11 and raised in Nepal. I am so
12
13
14 happy
```

CombinationSeparator.txt contains the file having combination of space, newline, and tabs. For example, in this file there are 7 new lines between words “born” and “and” which is supported per the user requirement.

```
assertEquals(correctResult, wordCount);
assertTrue(charCount==48 && lineCount==14);
```

6. For the input with Words, Number and separators combined.

[Test for Numbers]



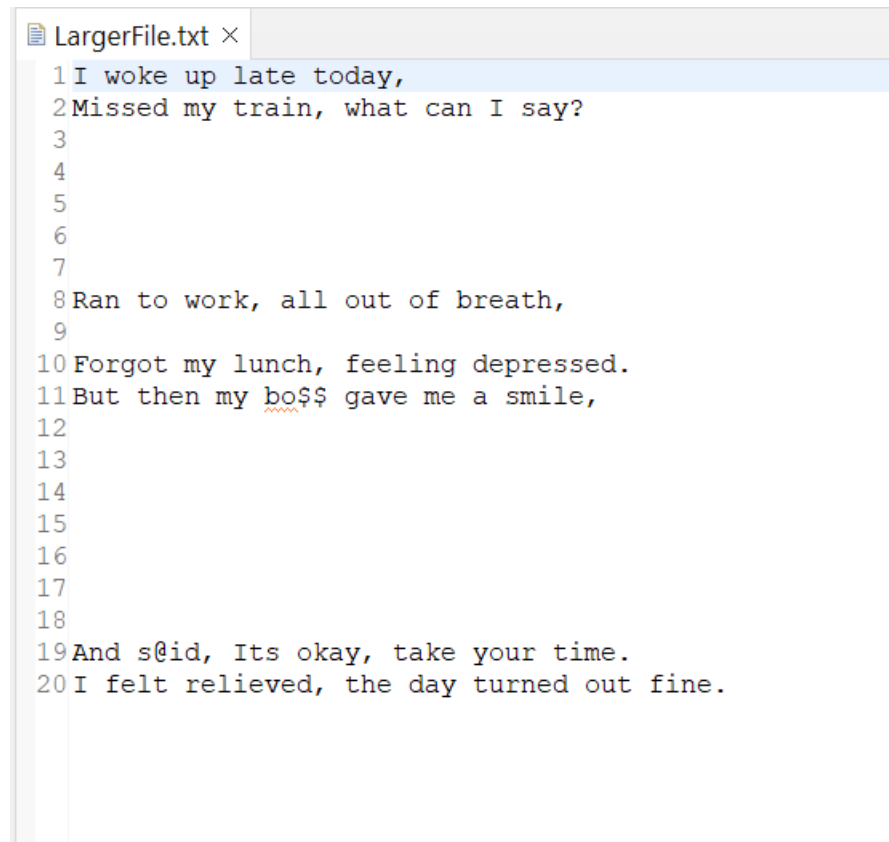
```
WordAndNumber.txt ×
1 123 is my favorite number I donot know why may be 23 is a number and not a word why I do not know why
```

WordAndNumber.txt contains words, numbers and separators. Each time the word or number is repeated, the frequency of wordcount increments. This file also has multiple separators combination which is correctly handled and supported.

```
assertEquals(correctResult, wordCount);
assertTrue(charCount==77 && lineCount==1);
```

7. For Larger Files with more characters.

[Test for Multiple word, multiple character, Multi Line, Multi Separator and Special characters]

A screenshot of a text editor window titled 'LargerFile.txt'. The editor contains 20 lines of text. Line 1: 'I woke up late today,'. Line 2: 'Missed my train, what can I say?'. Line 3: empty. Line 4: empty. Line 5: empty. Line 6: empty. Line 7: empty. Line 8: 'Ran to work, all out of breath,'. Line 9: empty. Line 10: 'Forgot my lunch, feeling depressed.'. Line 11: 'But then my bo\$\$ gave me a smile,'. Line 12: empty. Line 13: empty. Line 14: empty. Line 15: empty. Line 16: empty. Line 17: empty. Line 18: empty. Line 19: 'And s@id, Its okay, take your time.'. Line 20: 'I felt relieved, the day turned out fine.'.

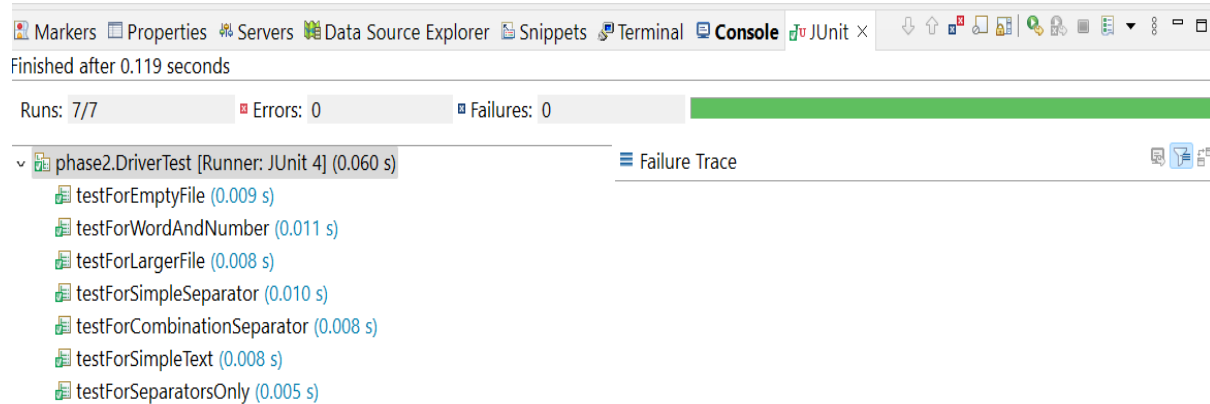
```
LargerFile.txt ×
1 I woke up late today,
2 Missed my train, what can I say?
3
4
5
6
7
8 Ran to work, all out of breath,
9
10 Forgot my lunch, feeling depressed.
11 But then my bo$$ gave me a smile,
12
13
14
15
16
17
18
19 And s@id, Its okay, take your time.
20 I felt relieved, the day turned out fine.
```

LargerFile.txt contains more words, characters, multiple separators and special characters. This test case is tested to check for special characters.

```
assertEquals(correctResult, wordCount);
assertTrue(charCount == 188 && lineCount == 20);
```

Hence, all the testcases above are tested and are **PASSED**

Testing on Eclipse IDE:



Testing on Command line:

```
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running phase1.WordCounterTest
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.138 s - in phase1.WordCounterTest
[INFO] Running phase2.DriverTest
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.021 s - in phase2.DriverTest
[INFO] Results:
[INFO] Tests run: 13, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- jar:3.3.0:jar (default-jar) @ WordCounter ---
[INFO] Building jar: C:\Users\samit\OneDrive\Desktop\CS5103_qkp232_SamitaShrestha\CS5103_qkp232_Samita\target\Wo
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.102 s
[INFO] Finished at: 2023-04-02T19:32:52-05:00
[INFO] -----
C:\Users\samit\OneDrive\Desktop\CS5103_qkp232_SamitaShrestha\CS5103_qkp232_Samita>
```

Sample Run:

1. SimpleText.txt

```
C:\Users\samit\OneDrive\Desktop\CS5103_qkp232_SamitaShrestha\CS5103_qkp232_Samita>java
-cp target\WordCounter-0.0.1-SNAPSHOT.jar phase2.Driver C:\Users\samit\OneDrive\Desktop
\CS5103_qkp232_SamitaShrestha\CS5103_qkp232_Samita\SimpleText.txt

The words and their frequency are:
+-----+
| Word | Frequency |
+-----+
| the  | 1 |
| with | 1 |
| Hello | 1 |
| words. | 1 |
| This | 2 |
| is   | 1 |
| simple | 1 |
| text | 1 |
| just | 3 |
| repeated | 2 |
+-----+

The number of lines are: 1
The number of characters are: 66

C:\Users\samit\OneDrive\Desktop\CS5103_qkp232_SamitaShrestha\CS5103_qkp232_Samita>
```

2. WordAndNumber.txt

```
C:\Users\samit\OneDrive\Desktop\CS5103_qkp232_SamitaShrestha\CS5103_qkp232_Samita>java -cp target\WordCounter-0.0.1-SNAPSHOT.jar phase2.Driver C:\Us
ers\samit\OneDrive\Desktop\CS5103_qkp232_SamitaShrestha\CS5103_qkp232_Sami
ta\WordAndNumber.txt

The words and their frequency are:
+-----+
| Word | Frequency |
+-----+
| 23 | 1 |
| a | 2 |
| be | 1 |
| may | 1 |
| I | 2 |
| why | 3 |
| is | 2 |
| do | 1 |
| my | 1 |
| number | 2 |
| 123 | 1 |
| not | 2 |
| and | 1 |
| know | 2 |
| favorite | 1 |
| donot | 1 |
| word | 1 |
+-----+

The number of lines are: 1
The number of characters are: 77
```

3. LargerFile.txt

```
C:\Users\samit\OneDrive\Desktop\CS5103_qkp232_SamitaShrestha\CS5103_qkp232_Samita>
xt
```

The words and their frequency are:

Word	Frequency
say?	1
train,	1
turned	1
woke	1
Forgot	1
relieved,	1
I	3
Its	1
smile,	1
your	1
out	2
depressed.	1
can	1
late	1
time.	1
okay,	1
of	1
s@id,	1
me	1
Missed	1
up	1
day	1
all	1
But	1
a	1
bo\$\$	1
breath,	1
felt	1
feeling	1
then	1
my	3
lunch,	1
the	1
take	1
what	1
today,	1
And	1
gave	1
work,	1
fine.	1
to	1
Ran	1

The number of lines are: 20

The number of characters are: 188

```
C:\Users\samit\OneDrive\Desktop\CS5103_qkp232_SamitaShrestha\CS5103_qkp232_Samita>
```

=====

Priority	Line	created	Rule	Error Message
▶	16	Sun A...	Sy...	SystemPrintl...
▶	26	Sun A...	Sy...	SystemPrintl...
▶	46	Sun A...	Sy...	SystemPrintl...
▶	45	Sun A...	Sy...	SystemPrintl...
▶	52	Sun A...	Sy...	SystemPrintl...
▶	43	Sun A...	Sy...	SystemPrintl...
▶	44	Sun A...	Sy...	SystemPrintl...
▶	51	Sun A...	Sy...	SystemPrintl...
▶	50	Sun A...	Sy...	SystemPrintl...
▶	36	Sun A...	Sy...	SystemPrintl...
▶	48	Sun A...	Sy...	SystemPrintl...