

CS4218 Rebuttal Report

(Team 17)

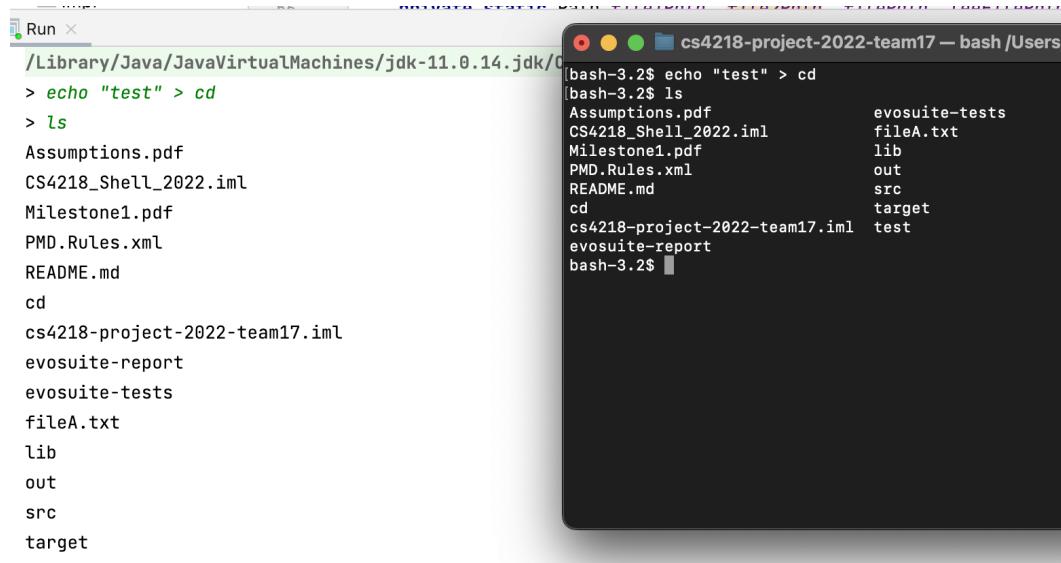
There are [references](#) to our reason for invalid bugs found below this table.

Bug Report Number	Status	Reason if invalid	Comments (this column should be left empty; for marking usage)
1.	Invalid	Working as intended. The behavior is the same as Linux where a file named "cd" will be created with "test" written into it. Otherwise, the directory does not move and no message is output.	
2.	Valid		
3.	Invalid	Working as intended. The output message is the same as Linux.	
4.	Invalid.	Working as intended. The team referenced Linux using ubuntu 18.04 and found that values with 0 are shown as well.	
5.	Invalid	Working as intended. We have tested it on Linux, the behavior of our implementation is the same as follows: 1. "grep: nonexistent: No such file or directory" is output in the shell 2. An empty file named fileA.txt is created	
6.	Valid		
7.	Valid		
8.	Invalid	Unable to replicate this bug. Our implementation did start on a new line.	
9.	Invalid	Working as intended as running cp with invalid files (assuming INVALID.txt is non-existent) is not supposed to create any folder. Our assumption never states how such a command creates a folder either	
10.	Invalid	Unable to replicate this bug. We have attempted to run the command with different content (pattern matched or no pattern matched), the output is the same as Linux implementation.	
11.	Invalid	Working as intended. The command <code>uniq -cd *.txt</code> here would be seen as <code>uniq -cd file1.txt file2.txt</code> after file resolution where	

		file1.txt is the input file and file2.txt is the output file. I believe that file2.txt's contents should be replaced as this is the behavior as described in the project description, since file2.txt here is the output file to write the uniq output to and there will not be any output on the console as output file is provided. This can also be seen in the Linux implementation. Hence, I believe our implementation of uniq is working as described and works the same as Linux's implementation.	
12.	Valid		
13.	Invalid	Based on Linux implementation, the given bug report test case command does not throw any exception.	
14.	Invalid	Unable to replicate this bug. Working as intended.	
15.	Valid		
16.	Valid		
17.	Valid		
18.	Invalid	The exact behavior for this situation is not stated in the project specification. According to Linux's implementation, there is no prompting of user input for the second "-" and it will simply read from the redirected input. What we have in our implementation is in accordance with Linux's behavior.	
19.	Invalid	There does not seem to be an "invalid input" error produced by Linux when sort "" command is provided, there does not seem to be an indication of this behavior mentioned on the project description as well. Also, our sort command throws an exception as expected, only a slight difference in the exception thrown compared to Linux's exception.	
20.	Valid		
21.	Invalid	Unable to replicate this bug on Windows or Linux, no matter whether a folder named "rando" exists or not. No file with erroneous name has been generated	
22.	Invalid	Unable to replicate this bug. Not sure of the contents of fileB.txt mentioned here. However, I have tried replicating this bug with the same command, a valid input file with "fileB.txt" as its name and some random contents in the input file. It seems to work as expected in Linux and on Windows.	

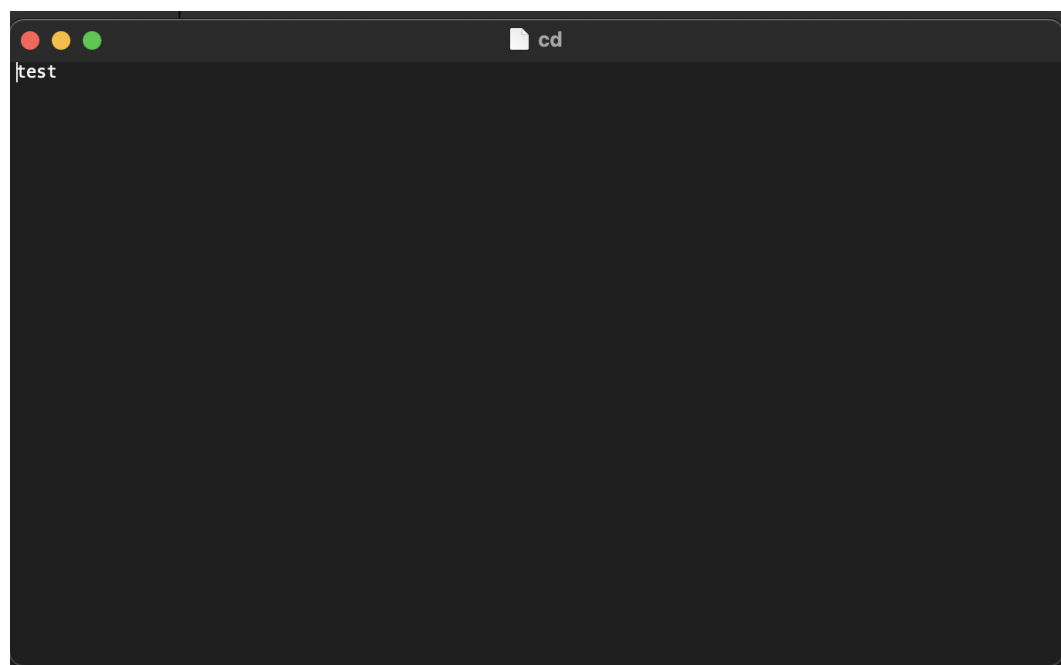
Evidence (if any)

Bug Report 1:



The screenshot shows a terminal window titled "cs4218-project-2022-team17 — bash /Users/". The user has entered the command `echo "test" > cd` and then `ls`. The output of `ls` is a list of files and directories: `Assumptions.pdf`, `CS4218_Shell_2022.iml`, `Milestone1.pdf`, `PMD.Rules.xml`, `README.md`, `cd`, `cs4218-project-2022-team17.iml`, `evosuite-report`, `evosuite-tests`, `fileA.txt`, `lib`, `out`, `src`, and `target`. The user has then entered `cd` to navigate into the `test` directory.

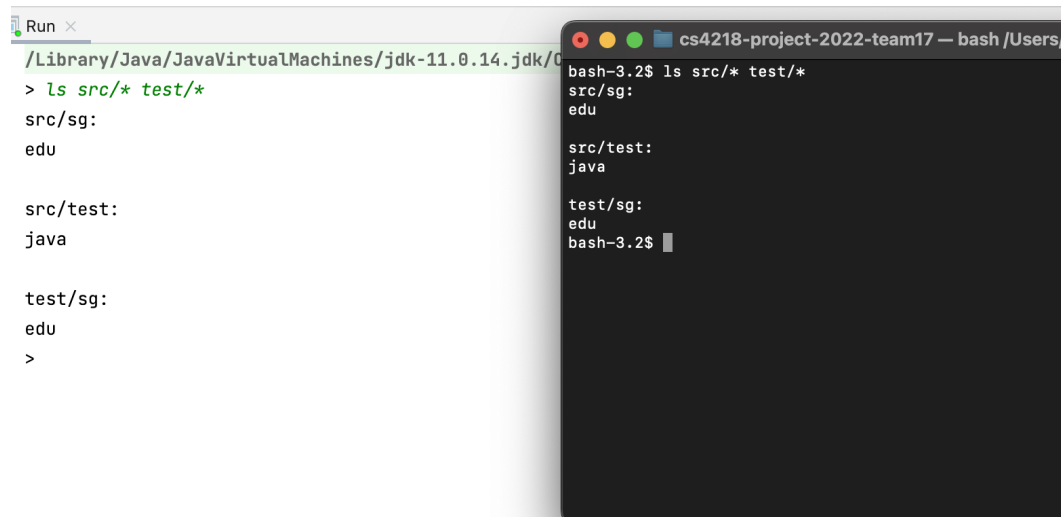
```
bash-3.2$ echo "test" > cd
bash-3.2$ ls
Assumptions.pdf          evosuite-tests
CS4218_Shell_2022.iml    fileA.txt
Milestone1.pdf           lib
PMD.Rules.xml            out
README.md                src
cd                        target
cs4218-project-2022-team17.iml test
evosuite-report
bash-3.2$
```



The screenshot shows a terminal window titled "cd". The user has entered the command `cd` and the output is `|test`, indicating that the current directory is `test`.

```
|test
```

Bug Report 3:



The image shows two terminal windows. The left window, titled 'Run', shows a directory listing for the path `/Library/Java/JavaVirtualMachines/jdk-11.0.14.jdk/Contents/Home`. The command `> ls src/* test/*` is entered, and the output is:

```
src/sg:
edu

src/test:
java

test/sg:
edu
>
```

The right window, titled 'cs4218-project-2022-team17 — bash /Users/...', shows a similar directory listing. The command `bash-3.2$ ls src/* test/*` is entered, and the output is:

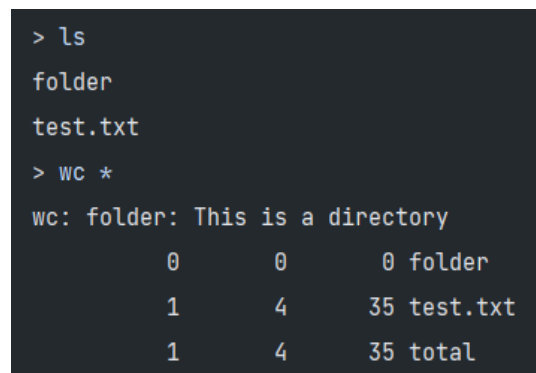
```
src/sg:
edu

src/test:
java

test/sg:
edu
bash-3.2$
```

Bug Report 4:

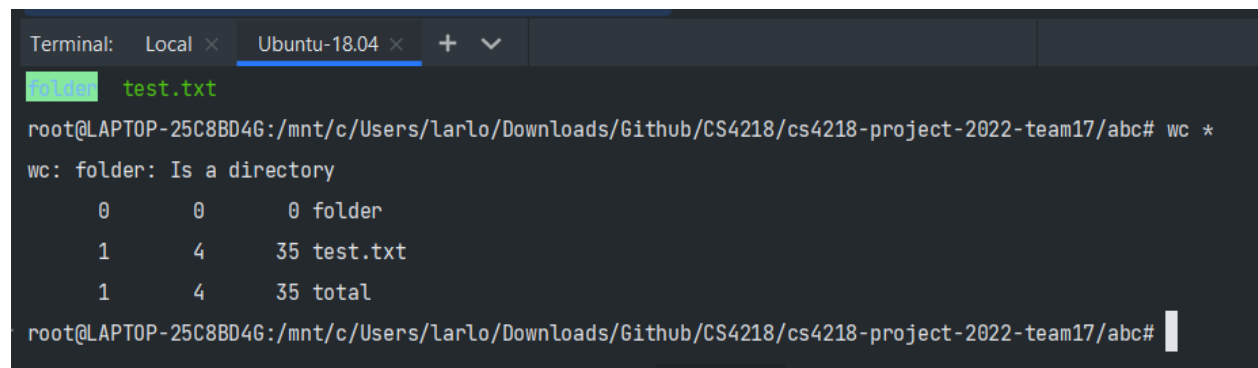
Actual Output:



The image shows a terminal window with the following output:

```
> ls
folder
test.txt
> wc *
wc: folder: This is a directory
      0      0      0 folder
      1      4     35 test.txt
      1      4     35 total
```

Output from Ubuntu 18.04 (Unix):

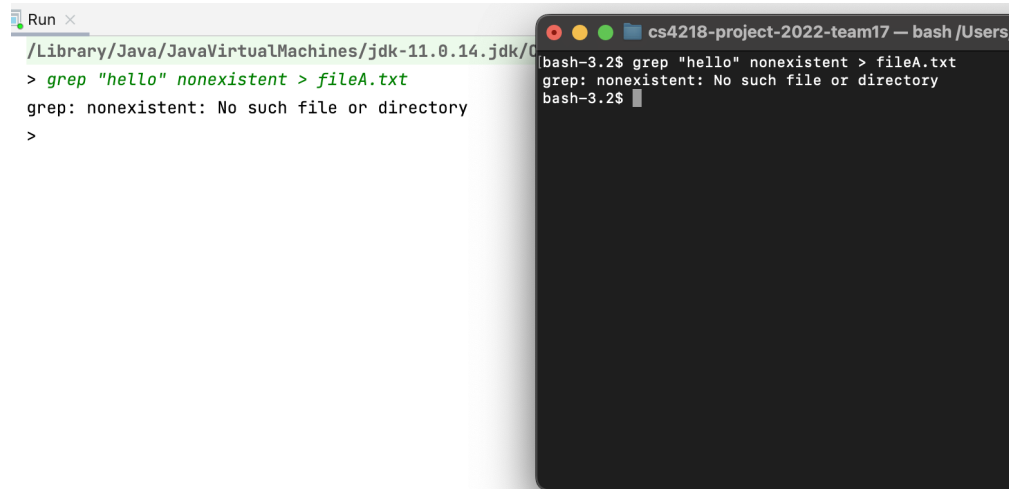


The image shows a terminal window titled 'Terminal: Local x Ubuntu-18.04 x + v'. The output is:

```
folder test.txt
root@LAPTOP-25C8BD46:/mnt/c/Users/Larlo/Downloads/Github/CS4218/cs4218-project-2022-team17/abc# wc *
wc: folder: Is a directory
      0      0      0 folder
      1      4     35 test.txt
      1      4     35 total
root@LAPTOP-25C8BD46:/mnt/c/Users/Larlo/Downloads/Github/CS4218/cs4218-project-2022-team17/abc#
```

Bug Report 5:

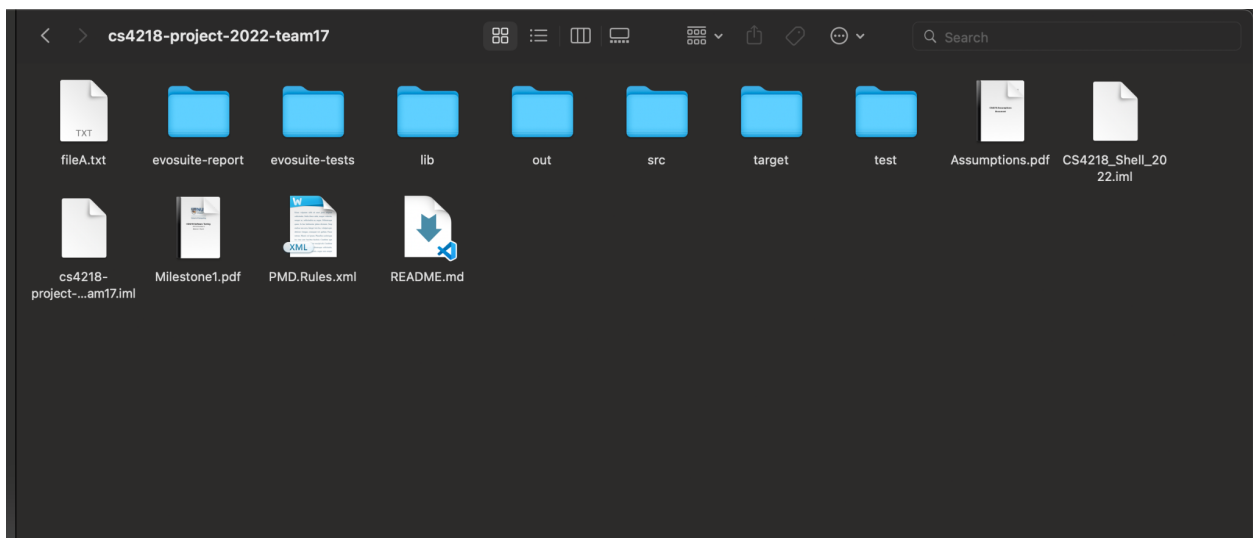
Output from our implementation and bash (identical)



The image shows two terminal windows side-by-side. The left window has a title bar 'Run' and shows the command `> grep "hello" nonexistent > fileA.txt` being executed. The output is `grep: nonexistent: No such file or directory`. The right window has a title bar 'cs4218-project-2022-team17 — bash /Users' and shows the same command and output in a bash shell.

```
Run
/Library/Java/JavaVirtualMachines/jdk-11.0.14.jdk/...
> grep "hello" nonexistent > fileA.txt
grep: nonexistent: No such file or directory
>

cs4218-project-2022-team17 — bash /Users
bash-3.2$ grep "hello" nonexistent > fileA.txt
grep: nonexistent: No such file or directory
bash-3.2$
```



fileA.txt is created as a result from the command on our implementation.

Bug Report 8:

```
> rm -r
> |
```

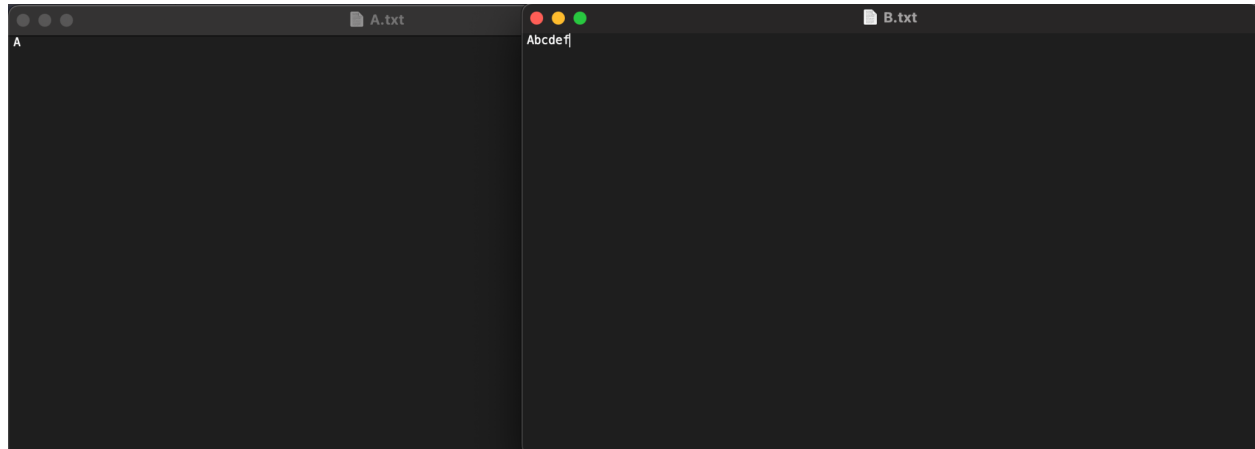
Bug Report 9:

Since the folder does not exist, it is expected for the application to throw a “no such directory”. In that case, our assumption never states that the invalid folder should be created

```
> cp pom.xml INVALID.txt README.md elephant
cp: D:\Git repositories\cs4218-project-2022-team17\elephant: No such directory
```

Bug Report 10:

File created for demonstration purpose: A.txt and B.txt



Our implementation ran on Mac:

```
> paste *.txt | grep "A"
A  Abcdef
> paste *.txt | grep A
A  Abcdef
> |
```

Bug Report 11:

Project description of uniq:

9.14. uniq

Description

The `uniq` command filters **adjacent** matching lines from `INPUT_FILE` (or standard input) and writes to an `OUTPUT_FILE` (or to standard output). With no options, matching lines are merged to the first occurrence.

Command format

```
uniq [Options] [INPUT_FILE [OUTPUT_FILE]]
```

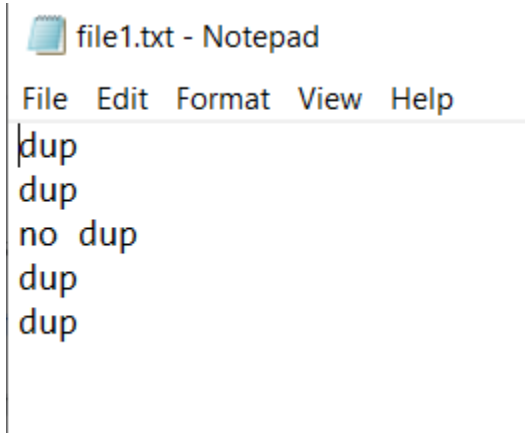
Options – One or more of the following options may appear:

- c – Prefix lines by the number of occurrences of **adjacent** duplicate lines
- d – Print only duplicate lines, **one for each group**
- D – Print all duplicate lines (takes precedence if used with d flag)

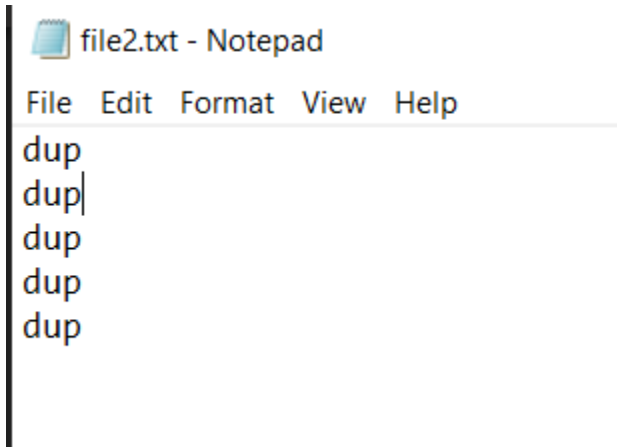
If `INPUT_FILE` is not specified or `INPUT_FILE` is '-', read standard input.

If `OUTPUT_FILE` is not specified, write to standard output.

Initial file1.txt content:



Initial file2.txt content:



Our implementation output ran on Windows:

```
> cd 'test dir'
> ls
file1.txt
file2.txt
> uniq -cd *.txt
> |
```

file2.txt content after running command on our implementation on Windows:



file2.txt - Notepad

File Edit Format View Help

2 dup

2 dup|

Our implementation out ran on Mac:

```
> ls
file1.txt
file2.txt
> uniq -cd *.txt
> |
```


file2.txt content after running command on our implementation on Mac:

file1.txt	file2.txt
1	2 dup
2	2 dup
3	

WSL Linux console output:


```
yeoutzer@LAPTOP-5K5LKMQR:/mnt/c/Users/Yeou Tzer/Desktop/Sch/CS4218/test dir$ ls
file1.txt  file2.txt
yeoutzer@LAPTOP-5K5LKMQR:/mnt/c/Users/Yeou Tzer/Desktop/Sch/CS4218/test dir$ uniq -cd *.txt
yeoutzer@LAPTOP-5K5LKMQR:/mnt/c/Users/Yeou Tzer/Desktop/Sch/CS4218/test dir$
```

file2.txt content after WSL Linux command:

 file2.txt - Notepad

File Edit Format View Help

```
|      2 dup
|      2 dup
```

(Seems to be working as described and works the same as on Linux)

Bug Report 13:

The given bug report tests case command does not throw an exception based on Linux implementation.

Linux/Mac Terminal:

```
bugreport % cat -n empty.txt > fileB.txt fileB.txt
bugreport %
```

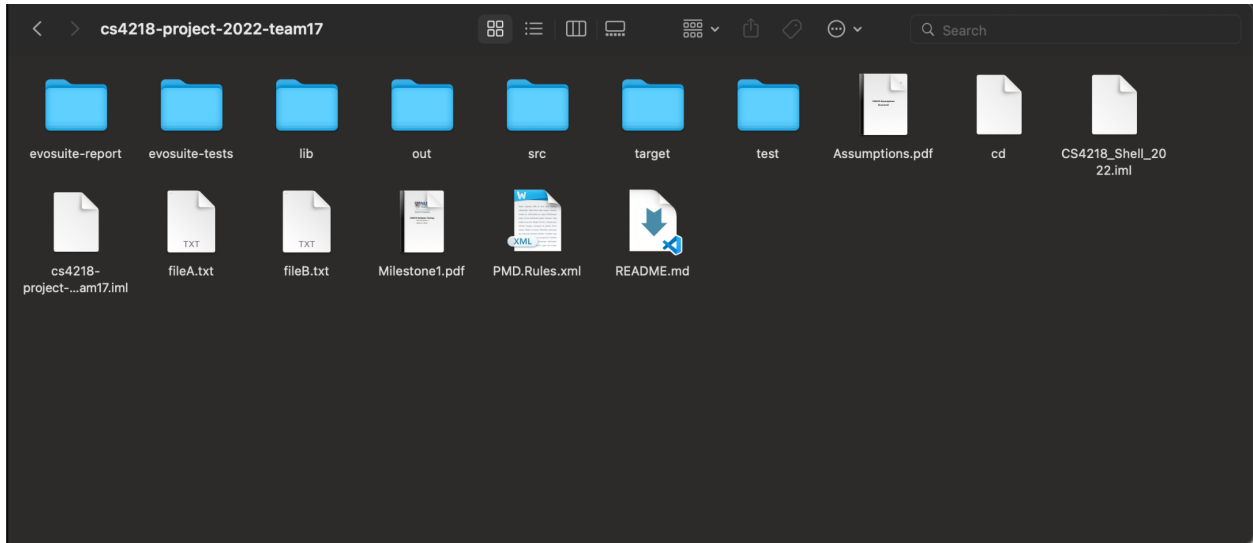
Our Implementation:

```
> cat -n empty.txt > fileB.txt fileB.txt
> |
```

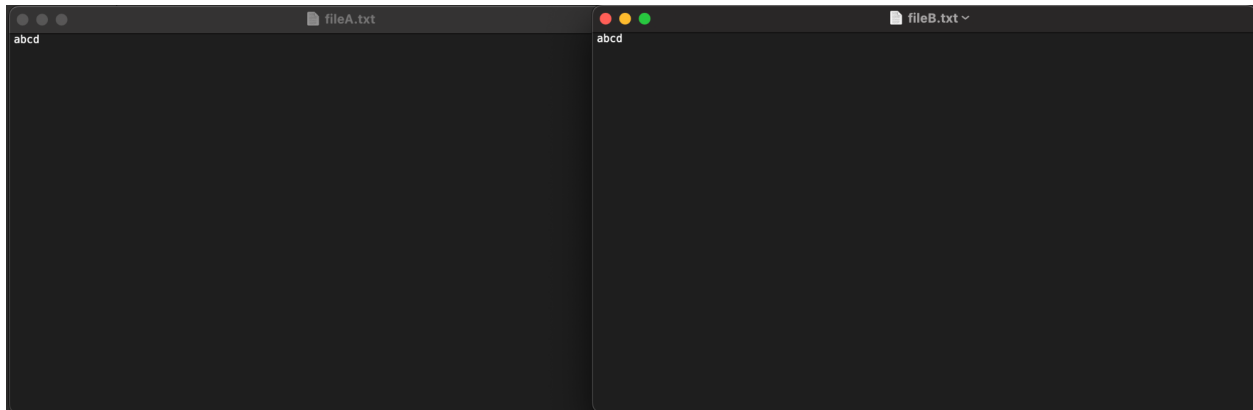
Bug Report 14:

```
> tee `ls *.txt`
abcd
abcd
^D
```

fileA.txt and fileB.txt is created for demonstration purpose



Output from tee written into both txt files as expected



Bug Report 15:

```
> ls
folder
test.txt
> mv test.txt folder\..
mv: Cannot copy a directory into itself
> mv test.txt folder/..  
mv: Cannot copy a directory into itself
> |
```

Bug Report 18:

Exact same output is produced when we run the same command on our application and on Linux:

Our implementation:

```
> paste - a.txt - < b.txt > output.txt
> cat output.txt
!  1  @
#  2  $
%  3  ^
&  4  *
(  5  )
-      +
```

Linux's implementation:

```
✦ paste - a.txt - < b.txt > output.txt
✦ cat output.txt
!  1  @
#  2  $
%  3  ^
&  4  *
(  5  )
-      +
```

Bug Report 19:

Our implementation output ran on Windows:

```
> sort ""
sort: : This is a directory
> |
```

Our implementation output ran on Mac:

```
> sort ""
sort: : This is a directory
```

WSL Linux Output:

```
yeoutzer@LAPTOP-5K5LKMQR:/mnt/c/Users/Yeou Tzer/Desktop/Sch/CS4218/test dir$ sort ""
sort: cannot read: '': No such file or directory
yeoutzer@LAPTOP-5K5LKMQR:/mnt/c/Users/Yeou Tzer/Desktop/Sch/CS4218/test dir$
```

(No such “invalid input” error thrown on Linux.)

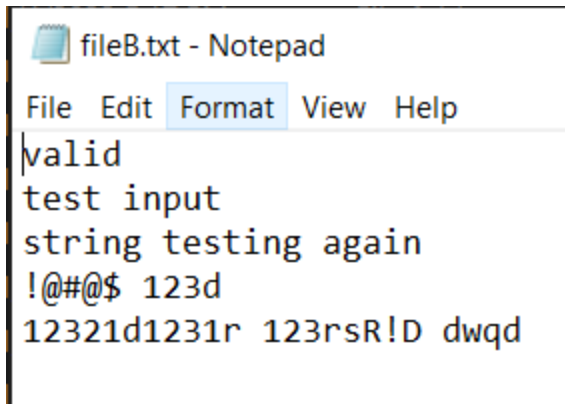
Bug Report 21:

Running the command will result in a folder named “rando” being created but no file with an erroneous name has been created

DIR	1/4/2022 5:57 pm	File folder	
evosuite	16/3/2022 10:20 am	File folder	
evosuite-report	20/3/2022 8:50 am	File folder	
evosuite-tests	18/3/2022 6:45 pm	File folder	
jacoco-resources	18/3/2022 4:02 pm	File folder	
out	11/2/2022 12:30 pm	File folder	
pit-report	1/4/2022 5:57 pm	File folder	
rando	1/4/2022 6:17 pm	File folder	
report	1/4/2022 5:57 pm	File folder	
src	27/2/2022 9:01 am	File folder	
test	10/3/2022 11:16 pm	File folder	
.gitignore	19/2/2022 3:47 pm	Text Document	3 KB
a	1/4/2022 6:02 pm	TXT File	1 KB
b	1/4/2022 6:02 pm	TXT File	1 KB
c	1/4/2022 6:02 pm	TXT File	1 KB
CS4218_Shell_2022.iml	7/2/2022 2:23 pm	IML File	1 KB
d	1/4/2022 6:01 pm	TXT File	0 KB
output	1/4/2022 6:03 pm	TXT File	1 KB
PMD.Rules	1/4/2022 5:57 pm	XML Document	11 KB
README	1/4/2022 5:57 pm	MD File	1 KB

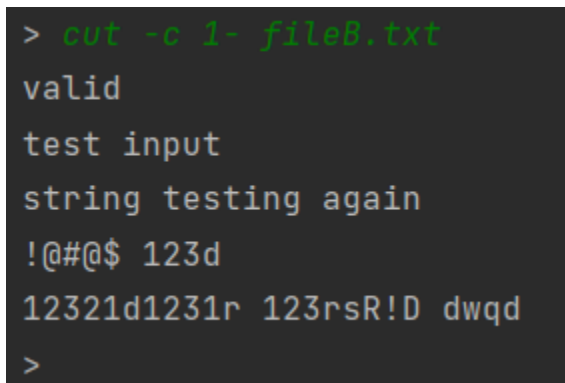
Bug Report 22:

File content used:



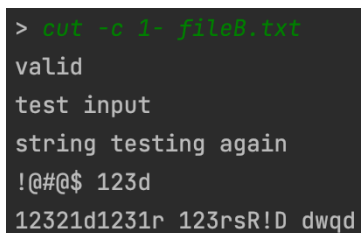
```
fileB.txt - Notepad
File Edit Format View Help
valid
test input
string testing again
!@#@$ 123d
12321d1231r 123rsR!D dwqd
```

Our implementation output ran on Windows:



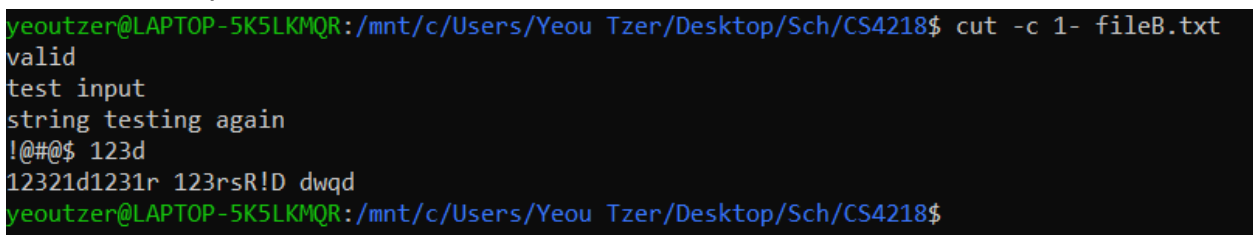
```
> cut -c 1- fileB.txt
valid
test input
string testing again
!@#@$ 123d
12321d1231r 123rsR!D dwqd
>
```

Our implementation output ran on Mac:



```
> cut -c 1- fileB.txt
valid
test input
string testing again
!@#@$ 123d
12321d1231r 123rsR!D dwqd
```

WSL Linux output:



```
yeoutzer@LAPTOP-5K5LKMQR:/mnt/c/Users/Yeou Tzer/Desktop/Sch/CS4218$ cut -c 1- fileB.txt
valid
test input
string testing again
!@#@$ 123d
12321d1231r 123rsR!D dwqd
yeoutzer@LAPTOP-5K5LKMQR:/mnt/c/Users/Yeou Tzer/Desktop/Sch/CS4218$
```

(Outputs are the same on both our implementation and on Linux. Unable to replicate bug.)