<u>Updated Rebuttal</u> (16 Valid Bugs out of 34 Reported)

Bug Report Number	Status (Valid / Invalid)	Reason if invalid	Comments (this column should be left empty; for marking usage)
1	Valid	The decision not to implement the multi-dash feature was due to the interest of time and the possibility of generating more bugs from the implementation. I have stress that, overall, our team did not skimp on our implementation. This was an exception, not the rule. We seek the teaching team's kind understanding and hope we will not be penalized too heavily on this.	
2	Valid		
3	Valid		
4	Invalid	We have tested using the steps to reproduce and unable to reproduce the error specified in the description with mv test.txt test2.txt where test.txt is a 500MB file. We discovered issues with the bug report 4 steps to reproduce, the step "[Powershell] fsutil file createnew test.txt 52428800" creates a test.txt of 50MB and not 500MB as 500MB is 524288000 not 52428800 as stated in the bug report in many areas involving 500MB mentioned in the report. When we reproduce the steps for both 50MB and 500MB, we are able to observe the expected output which is the renaming of the dummy test.txt file to test2.txt file using the mv command (mv test1.txt test2.txt) Hence, due to inability to reproduce the bug, this bug is considered invalid.	
5	Invalid	We discovered issues with the bug report 4 steps to reproduce, the step "[Powershell] fsutil file createnew test.txt 52428800" creates a test.txt of 50MB and not 500MB as 500MB is 524288000 not 52428800 as	

		stated in the bug report in many areas involving 500MB mentioned in the report. Moreover, while we appreciate the team's effort in stress testing our system, it is important to note that fsutils creates a file with a stated memory size by filling it with the NUL character. We have tried to open the file created on notepad, as well as notepad++, even these applications hang or have trouble reading. This is because of the contents of the file, consisting purely of NUL characters. When the commands provided are executed on linux, even linux's paste hangs. Note that, in this project, optimisation is not factored in. We are not concerned about ensuring the program can run fast enough, or look into the memory aspects of heap. Basically, this is out of scope of this project and module, which is why our team did not download a 1GB text file to stress test Team S' system as well.	
6	Valid		
7	Valid		
8	Valid	This is the exception thrown when the file failed to move for method Files.move. It is the intended behaviour to throw the exception that the files move failed when attempting to move to the same location.	
9	Valid		
10	Invalid	Duplicated Bug regarding file path as Bug 9	
11	Valid		
12	Invalid	Duplicated Bug regarding file path as Bug 9	
13	Valid (with the exception of replacement index 2, that is not valid)	Not completely valid. Please see the following counter example: Command: sed 's/ok/replace/2' okok.txt File content of okok.txt: ok ok ok ok Actual Output: ok replace ok ok However, we do accept that from the 3rd replacement index onwards, it does not replace.	

14	Valid	Special Symbols like ?, have their own special meanings. We have to note that our shell project attempts to replicate some of the core behaviors of linux with the project specifications in mind, but to cover every special symbol, and its related special meanings/functions, is beyond the limited scope of this project. This is implicit, as even our assumptions cannot foresee the multitude of possibilities and special functions that linux has to offer and provide assumptions for it. It is not realistic, and the testing team's assumptions report do not do that as well.	
15	Invalid	The idea is the same as in rebuttal 14. Again, the special symbol '?' is used, which according to linux, "represent any single character". The explanation is the same. Special symbols are not considered in our implementation. Please see rebuttal 14 for more details.	
16	Invalid	Again, the special symbol '?' is used. Please refer to rebuttal 15, followed by rebuttal 14. While we appreciate the team's effort in testing our system, such "bugs" are beyond the scope of this project.	
17	Invalid	Please refer to the Luminus forum post titled "diff application questions". Dr Cristina has explained that *"Edited: When comparing two folders, you need to compare the names of the sub-folders, and the name of the files and content for files with the same name. For text files, show the differences, while for non-text files show only if they differ or not. "* In this context, we see that the scope of diff has been limited to one level down, meaning one subfolder. With that in mind, I printed only one level up from the filename.	
18	Valid	This is a valid bug (albeit minor) as a duplicate output line was printed.	
19	invalid (but can be combined	This is a valid bug (can be categorized with number 18 as the same bug) as the identical message is not printed.	

	with 18 as both are similar)		
20	Invalid	Please see rebuttal 21. 20 and 21 are essentially the same issue.	
21	Invalid	On face value, if we were to follow the project specifications, as opposed to purely Linux, it does not explicitly show that when 2 folders are the same, we print nothing. The objective of printing out the other contents, is to show the user what are the similarities and differences (no difference for this case) between the folders. More details, even though they may be completely the same, is harmless to the user. We are more concerned about cases where folders are very similar and have just 1 difference. This is where printing out the other information becomes helpful.	
22	Valid	In general, “bugs” 22, 24, 25 are about exception handling. These are not valid bugs as we have scaled down our exception handling due to the scope of this project. Messages are generalised to ensure broader meaning. For instance, “invalid argument” is used in diff. Please note that this is not the only exception message we have, but this is the message that the testing team has encountered given their test cases. A general exception message is to cover all possible exceptions, without going to the specifics. To tackle and address every possible exception for every one of our applications is not feasible, given the project constraints and limitations in terms of time. We seek the kind understanding of the teaching team as well as the testing team on this. Else, all our testing will end up being about exceptions due to the sheer amount of possible events and exception messages that could occur.	
23	Invalid	Multi-dash not supported. For diff’s case, multi-dash is different from paste. The added value is minimal, as it prints nothing because both dashes use contents from the same file in stdin. I	

		could easily add in a check for 2 dashes, and print nothing. However, I felt that standardising with paste makes the system more consistent, since paste also does not support multi dash for our implementation. This also prevents confusion for the user, particularly since consistency in a software product is key, to making it understandable for the user.	
24	Invalid	See rebuttal 22	
25	Invalid	See rebuttal 22	
26	Invalid	Error message is thrown with intended reasoning and the reason in exception thrown is correct.	
27	Valid	For Windows the bug is present. For Unix based system it is working as intended.	
28	Invalid	It is possible to use the command ls -R -d and the output is still correct as in line with shell. The assumption is wrongly phrased due to misunderstanding of project specifications as ls -R -d and ls -d ‘s results are the same but the program works as it is. The way unix shell works if check unix shell it is the same in project specification in terms of the flags and their definition of the flag so basically the -d flag if referring to manpage and project specification list directories themselves and not their contents (http://man7.org/linux/man-pages/man1/ls.1.html) while -R list recursively their subdirectories. Since the contents are not listed the output should only be the folder name itself. Input : “ls -d -R src” will just output “src” which is similar to “ls -d”.	
29	Invalid	Error message is intended to direct that the flag provided should be -name.	
30	Invalid	Decreasing range for cut command is supposed to work as intended as executed in Windows CMD and Mac Terminal since project specification did not specify that decreasing range should throw an exception. In Terminal and	

		Windows CMD, decreasing range for cut command will print into the console based on the number of empty lines in the files or input stream instead of print an exception message. Furthermore, I did not disallow decreasing range for cut command in our Shell.	
31	Invalid	Duplicate of #11. I choose #11 to be the valid bug instead of #31 as #11 provide more details of the bugs instead of #31.	
32	Valid		
33	Valid	Explained in the assumptions of Paste.	
34	Valid	This was explicitly addressed in our assumptions report under grep. This was an intentional logic. Key reason for rejecting “-ic” is that the user might have a typo, accidentally adding the dash, when they wanted to grep “ic” from the file. I considered the fact that in Singapore, IC refers to identification cards. If IC does not have any significant meaning, we wouldn’t have rejected “-ic” as a valid flag. Although some may argue for that in Linux combining flags is common. This is our design choice. Just an additional logic based on what we feel is suitable for the user. We are not exactly following Linux’s behavior to the dot. The idea here is that it is better to split up the flags, to prevent situations like typos which could lead to different results. Furthermore, unlike in Diff, the flag -sBq, even without the dash “sBq, Bsq, qsB” means nothing significant.	