

As the bug report mentioned, they tested our codes using Linux/macOS shell implementation standards. We think this is incorrect. Since we are required to implement a shell based on our project description. Our codes should be tested on whether it can fulfill the requirements stated on the project description, but not be tested on whether it works the same as Linux/macOS shell.

Moreover, we noticed that teamX, one of the teams in our hackathon, claimed they follow the Unix shell standard in their assumption, so we tested their code in Unix standard as they claimed. But we never claimed we strictly followed the Unix shell standard.

Bug Report Number	Status	Reason if invalid	Comments (this column should be left empty; for marking usage)
1	Invalid	Explained above	
2	Valid		
3	Invalid	The command prompt is not a project requirement at all. Also, we never said that IS a command prompt and it should behave like a command prompt in the Assumptions.pdf. We do it just for visual assistance.	
4	invalid	Maybe we didn't make it clear enough in the assumption, we implemented the paste according to the example here <a href="https://www.geeksforgeeks.org/paste-command-in-linux-with-examples/">https://www.geeksforgeeks.org/paste-command-in-linux-with-examples/</a> This is discussed on the forum on 9 Feb 2020, thread "Paste Application". All the "-" read from the same stdin without copying it, that is, each "-" will read content from a new line rather than repeating what the last "-" read.	
5	Invalid	As mentioned in the project description, in the first line of the first paragraph of each application, it suggested that if no file is specified the application (both wc and sort) will take the stdin as the input. So, the bug description is due to being unable to send EOF to a java program. I think they didn't read the project description carefully.	
6	Valid	But the shell just exits with 0 code, not crash.	
7	Valid	We should check whether the source file exists. We did throw an exception, but method run catches the exception and writes the exception message to stdout.	

8	Invalid	The cp does not have -r flag and also does not follow normal UNIX behaviour. We assume that copying a folder to a file will replace this file. We have the relative test but forget to add it in assumption.	
9	Invalid	We do not use the "/" at the tail to distinguish whether the destination is a file or folder. If the destination is an existing folder, then copy the file to this folder. Otherwise, we will copy the file to another file. This is our assumption but forget to add it.	
10	Valid		
11	Valid		
12	invalid	We think their expected behaviour of find: "Should output permission denied errors for relevant folders and proper output for valid folders. " is just an assumption made in their own implementation. In our implementation, we just throw a permission denied exception when `find` in an unreadable folder. So we see this as a difference of assumptions rather than a bug.	
13	invalid	We think this is not a correct test, according to project description page 19, 9.14. find, the command format for `find` is <i>find FOLDERS -name [filename]</i> where FOLDERS should be the name of the folder or folders. However, their test case was <i>find ls\1.txt -name '1.txt'</i> where they put a file name instead of a folder name as the first argument, and our application outputs the error message as intended.	
14	invalid	We didn't state clearly in the assumption, we ignore invalid flags for sort, we think this is acceptable behaviour and should not be considered as a bug. We have a test case that defines our expected behaviour ever since MS1, which is written in <code>testRunWithWrongSuffix()</code> in <code>src/test/sg/edu/nus/comp/cs4218/impl/app/SortApplicationTest.java</code> . We'll just add the assumption in MS3.	
15	invalid	We don't examine the correctness of the	

		<p>behaviour of our `sort` application based on the behaviour of Linux/macOS shell, so we didn't handle negative values.</p> <p>We consider "-" as a special character and the rule we used for sort is as in the project description page 19, 9.13. sort, in description "Compares each line character by character. A special character (e.g., +) comes before numbers. A number comes before capital letters. A capital letter comes before small letters, etc. Within each character class, the characters are sorted according to their ASCII value."</p>	
16	Invalid	We did throw an error to indicate "move" failed. Because normal "mv" do not have output.	
17	Invalid	We did throw an error to indicate "move" failed. Because normal "mv" do not have output.	
18	Valid	We miss understanding the -n flag of "mv".	
19	Invalid	We did throw an error to indicate "move" failed. Because normal "mv" do not have output.	
20	Invalid	It's true, but it is redundant with 21. We consider this bug as making up numbers.	
21	Valid		
22	Valid		
23	Invalid	We made an assumption about this situation, no matter what order is given, the output is the same. It is our fault that we did not complete the sentence, but we think the meaning of our assumption is obvious.	
24	Invalid	<p>We implemented the `cut` application following the specification in Q&amp;A, 9 Feb 2020, 1. "LIST - can be a list of two numbers separated by comma, a range of numbers or a single number."</p> <p>Input like "2-" or "-2" or "2," or ",2" are considered as invalid in our implementation.</p>	
25	Invalid	Same as No.24	
26	Invalid	In our implementation, a new line will be	

		replaced by a space in command substitution. Such behaviour is defined in the project description, page 12, 8.8. Command substitution, in Semantics.2. “Since our shell does not support multi-line commands, newlines in OUT should be replaced with spaces”.	
--	--	--	--