

Notes on Testing

- We generally followed Linux/macOS shell implementation standards for testing, unless explicitly stated in the assumption/project requirements(Q&A doc and project_description pdf).
- Bug tests provided will use official errors as defined in the project. However, outputs were checked manually to see if custom errors provided conveyed the same meaning. If they are not representative, we classify them as bugs.

Bugs Found

No.	Description	Testcase	Comments (this column should be left empty; for marking usage)
1	For all commands (except paste), an extra newline is printed after the output. Since this doesn't fit the UNIX shell standard, and wasn't explained in their assumption, we consider this as a bug.	'ls' 'cd src'	
2	If a folder has a number as its name, ls [folder] will not work	'ls 0'	
3	The command prompt doesn't show which directory you are in. It always stays the same.	-	
4	Wrong output for paste with multiple files and stdin	pasteMultipleFilesAndStdin	
5	Some commands with no arguments will cause the shell to stop taking in new commands (no args error not handled)	'wc' 'sort'	
6	cd with no arguments will crash the app with out of bounds exception	'cd'	
7	Exception not thrown for cp when source file doesn't exist. After being run, destination file is incorrectly deleted.	cpNonExistentSource	
8	Exception not thrown for copying directories	cpSourceIsDirectory	

	into files. Instead of throwing exception, the destination file is changed into a directory. This behaviour does not follow normal UNIX shell behaviour and was not documented in the assumptions, so is considered a bug		
9	Treats the non-existent folder “nonexistent/” as a file and creates it rather than handling qwerty as a directory(either by creating new directory called qwerty or throwing exception)	cpIntoNonExistentDirectory	
10	ls fails to return the correct error when used on folders with no permission. Breaks -R and causes shell to terminate with NullPointerException.	IsNoPermissions	
11	ls fails to provide correct output when used recursively on arguments <a> and where is a folder in <a>. May seem incorrectly formatted but may also be difficult to distinguish.	IsFolderWithinFolder	
12	find fails to execute on valid folder paths when folder with no read permission is provided. Should output permission denied errors for relevant folders and proper output for valid folders.	findMixedPermissions	
13	find incorrectly throws a file not found error and fails to deliver correct output when used on a valid file with regex argument containing the valid file's name. Should output the valid file's name.	findValidFile	
14	sort fails to return proper error for invalid flag - error claims no such file or directory.	sortInvalidArgs	
15	sort fails to properly sort numerically (-n). Linux/macOS shell compares using numerical chunks (from left to right) on each line and follow this ordering (smallest to largest):	sortNumerical	

	<ol style="list-style-type: none"> 1. Negative real numbers until -1 2. Non alphabetic characters (not numerical) 3. 0 4. Capitalised alphabetic characters (not numerical) 5. Non-capitalised alphabetic characters (not numerical) 6. Real numbers from 1 onwards 		
16	mv fails to return proper error for invalid file argument - error only prints a file path. Error should inform that file is not found.	mvInvalidFile	
17	mv fails to return proper error for moving valid folder into a valid file <a> - error states file already exists. Error should inform that <a> is not a directory. Was tested with both -n flag and no -n flag with reference to assumptions.	mvValidFolderIntoValidFile	
18	mv fails to replace valid file <a> over valid file with -n flag (according to assumptions). Does not replace without -n flag either. Should replace with no error.	mvReplaceValidFile	
19	mv fails to execute on valid file and folder paths arguments when file/folder paths arguments with no read permission are provided. Also fails to throw proper errors - errors only state a path. Should output permission denied errors for relevant files/folders and execute otherwise.	mvReplaceMixedFiles	
20	diff doesn't follow format shown in project description pdf example for comparing binary files(with -q flag): "Binary files A/image2.bmp and B/image2.bmp differ". Same for comparing binary files of the same name in two directories	diffComparingBinaryFilesOutputFormat	
21	diff doesn't follow behavior stated in Q&A for	diffComparingBinaryFi	

	comparing binary files: for non-text files show only if they differ or not. Same for comparing binary files of the same name in two directories	les	
22	Unexpected file not found exception thrown for diff comparing 2 directories with directories inside them	diffComparingDirectoriesWithDirectoriesInside	
23	Order not preserved of file and stdin in the output. (assumptions incomplete so we treat this behavior as bug)	diffFileWithStdin	
24	Cut: does not handle one integer range argument: eg: 2- or -2 Expected behaviour: Integer comes before -: substring starting from index of integer Integer comes after -: substring from start, ending at integer	oneIndexRangeArgument	
25	Cut: does not handle one integer list argument: eg: 2- or -2 Expected behaviour: Invalid range exception thrown	oneIndexListArgument	
26	Command Substitution: For every nested backquoted command, it prints extra space	nestedQuotingTest	