TEAM F:

All tests are done through the shell directly, i.e. using **ShellImpl**. A basic structure of files, folders and subfolders are provided in Structure.zip, and can be re-extracted should a test case affect the structure. It should be **extracted to the project root**.

The structure of the files and folders of the zip are as follows:

File1.txt
File2.txt
Main1
    ↳File1.txt
    ↳File2.txt
    ↳a.bmp
    ↳Sub1
        ↳a.bmp
        ↳b.bmp
        ↳File1.txt
        ↳File2.txt
    ↳Sub2
Main2
Main3
    ↳Sub3

This means there's File1.txt and File2.txt at the root, main1 and sub1. Subfolders sub1 and sub2 are in main1. Main2, main3, sub2 and sub3 are empty directories. All File2.txt are empty text files. Both a.bmp are the same, b.bmp is different.

In general, testing was done through using our domain knowledge, utilizing known problematic test cases, testing with edge cases in commands and checking for bugs that existed in the source code originally provided to us. Additionally, we read through the assumptions to look for any potential problems or faults that might contradict with the specification.

Bugs found:

Bug Report number	Description	Testcase	Comments
1	Paste does not support more than one '-'. While an assumption was	paste - -	

	made for this, the requirements (as clarified in the FAQ) is that it must support multiple '-' (as per 22 Feb 2020 clarification)		
2	If ls is executed on an empty directory, an empty line is printed. In the Linux implementation, nothing is printed.	ls main2	
3	cd main3/`ls main3` does not move the current working directory to main3/sub3. While an assumption is made for this, the specification does state that command substitution is performed after command-level parsing but before argument splitting, which heavily implies that this functionality should work. Additionally, the documentation in argument resolver (*e.g. arg: abc`1 2 3`xyz`4 5 6` (contents in `` is after command sub), expected: [abc1, 2, 3xyz4, 5, 6]* basically says that this should work.	cd main3/`ls main3`	
4	mv does not move large files, tested file with size 500MB and it just errors out with the name of the file as an error	[Powershell] fsutil file createnew test.txt 52428800 mv test.txt test2.txt	
5	Program crashes on attempting to read large files	[Powershell] fsutil file createnew test.txt 52428800 paste test.txt	
6	ArgumentResolver does not work correctly with backticks within double quotes, expecting an error with missing closing backtick.	echo "a`a"	
7	mv does not tell you that you're moving a folder into the same location. This is expected in both Windows Powershell and Unix.	mv main3 .	

8	mv on a parent folder to its child is caught through a catch-all generic exception	mv main3 main3/sub3	
9	mv does not check for current directory and does not tell user that mv failed to move a file	cd main1; mv file1.txt sub1/file3.txt	
10	mv does not check for current directory and does not tell user that mv failed to move a folder	cd main1; mv file1.txt file3.txt	
11	cut with an invalid list that includes both position and range values should have a proper error, but only says ‘For input string: “5-7”’. This is a default exception message from NumberFormatException and is not handled, and is merely caught by a catch all exception.	cut -b 1,3-5 file1.txt	
12	mv does not correctly identify when a file is moved to its own location with relative path	mv a.txt ../projFolder	
13	sed’s Xth match only works when X = 1, it does not work for X = 2, 3, 4...	echo no effort no hope | sed s/no/got/2	
14	sed does not accept valid regex expressions, input containing special regex characters are not accepted. The command should work as it does on Windows Bash and Linux.	echo valid regex does not work? | sed s/?/./	
15	sed replaces ‘perempuan?’ with ‘lelaki.?’ erroneously.	echo dia perempuan? | sed s/perempuan?/lelaki./	
16	grep does not accept valid regex expressions, input containing special regex characters are not accepted. The command should work as it does on Windows Bash and Linux.	echo hello world? | grep ?	

17	Diff: “Files main1/file1.txt and main1/sub1/file1.txt are identical” or something similar is expected. Instead, “Files main1/a.txt sub1/a.txt are identical” is given as output. The path for the second directory is incorrect.	diff main1/file1.txt main1/sub1/file1.txt -s	
18	Diff: Expected output: “Binary files main1/a.bmp and main1/sub1/b.bmp differ” Actual output: “Binary files main1/a.bmp sub1/b.bmp differ Binary files main1/a.bmp sub1/b.bmp differ” The contents of b.bmp are also lost.	diff main1/a.bmp main1/sub1/b.bmp	
19	Diff: Expected output: “Files main1/a.bmp main1/sub1/a.bmp are identical” Actual output: “Binary files main1/a.bmp sub1/a.bmp differ Binary files main1/a.bmp sub1/a.bmp differr” The contents of a.bmp in sub1 are also lost.	diff main1/a.bmp main1/sub1/a.bmp -s	
20	Diff: Expected output: “” Actual output: “Common subdirectories: main3/sub3 and main3/sub3”	diff main3 main3	

21	Diff: Expected output: “” (since -s only applies to FILES as specified in project brief) Actual output: “Files main1/a.bmp main1/a.bmp are identical Files … … are identical Common subdirectories: … and ...”	diff main1 main1 -s	
22	Diff: Expected output: “main4 does not exist” or something similar Actual output: “diff: Invalid arguments”. The arguments are not in an invalid format.	diff main1 main4	
23	Diff: Expected output: “” since it is always the same. Actual output: “diff: Invalid arguments”. The arguments are not in an invalid format.	diff - - < a.txt	
24	Diff: Expected output: “No such file or directory” or something similar, Actual output: “diff: Invalid arguments”. The arguments are not in an invalid format.	diff - - < file3.txt	

25	Diff: Expected output: “diff: main3/file1.txt: No such file or directory” Or if it is assumed that you cannot compare directory and file (not stated in assumptions), “diff: cannot compare directory with file” or something similar. Actual output: “diff: Invalid arguments”.	diff main3 file1.txt	
26	Ls: Expected output: “ls: cannot access ‘main4’: No such file or directory main4: ...” Actual output: “ls: No file or folder specified found”. Should list directories and files which exist and throw individual error message for those which don’t. See Windows Git Bash implementation.	ls main3 main4	
27	Ls: expected to work, the command provided is an example in the project brief, instead, the command fails with the error message: “Illegal char <*> at index”	ls -d */	
28	Ls: in the project brief, the command format was stated as such, “ls [-d][FOLDERS][-R]”. However, the team made an assumption that “ls -d -R FOLDERS is invalid”. This is contradictory. There is no	ls -d main1 -R	

	explanation why the command did not have the ability to list directories only recursively, which is definitely a legal use-case.		
29	Find: expected output: “find: invalid syntax” or something similar, actual output: “find: Flag must be -name”	find main1 -name hi -z	
30	cut decreasing range is not caught and outputs an empty line	cut -b 3-1 file1.txt	
31	cut with any invalid list should produce a proper error, but only says ‘For input string: “XX”’, where XX is the invalid string. This is a default exception message from NumberFormatException and is not handled, and is merely caught by a catch all exception.	cut -b 1, file1.txt cut -b a file1.txt cut -b 3-x file1.txt cut -b 1,2,3,4 file1.txt	
32	Globbing */* should expand to all files in subdirectories in the current directory	echo */*	

Potential Bugs found:

These are bugs that do not follow the typical unix convention and instead have been implemented differently, without a seemingly good reason for such an implementation. While the specification given in the project does not explicitly say that these are bugs, we feel that the specifications do imply that they should be considered as bugs. While these are also explained away in assumptions, we feel that without a sufficiently good reason for it everything in the list below should be considered bugs.

Bug Report number	Description	Testcase	Comments
33	If ‘-’ is supplied to paste as an argument, the contents of the standard input will appear before any contents in files,	paste a.txt -	

	even if the dash is the 2nd argument or later.		
34	grep does not accept “-ic”. An assumption was made that it is “not acceptable”, but given that many commands in the project brief allow option combination, and that it also works for grep in Linux and Windows, this should be enabled to for consistency and ease-of-use sake.	echo hello | grep -ic he	