Bug #	Description (What's wrong?)	Test Case	Type	Comments (for grading)
1	Executing any unsupported application throws exception and print the stacktrace instead of printing a meaningful error message.	`abcd`	general	

2	Executing any valid application but with invalid syntax throws exception and print the stacktrace instead of printing a meaningful error message
	
	For example,
	grep throws java.lang.ArrayIndexOutOfBoundsException: 0
	`paste` prints null
	`grep ++ pom.xml` throws java.util.regex.PatternSyntaxException
		`grep`
		`or`
		`paste`
		`or`
		`grep ++ pom.xml`
			general	

3	Executing any valid application but with invalid files or folders throws exception and print the stacktrace. The shell should have checked if the file or folder exist before executing the applications and print a meaningful error message if the file or folder does not exist. For example, grep invalid files throws `java.nio.file.NoSuchFileException` instead of meaningful error message	`grep <non-existent file>`	general	
4	Globbing returns results as if we are in the project directory instead of the actual current directory.	`cd /` `echo *`	general	
5	Sequence commands with trailing semicolon and whitespaces result in exception. (because the team's implementation of shell interpreted the trailing whitespace as an invalid app)	`ls;<space>`	general	
6	When consecutive semicolon is used, an invalid app exception thrown instead of invalid syntax (because the team's implementation of shell interpreted the empty string between the semicolons as an invalid app)	`ls;;`	general	

7	For sequence commands, if an exception is thrown during the execution of the first command, the execution of the second command should continue and may return a non-zero exit code. However, the team's implementation of the shell does not follow the above requirement.	`<invalid app>;ls`
		`or`
		`ls;<invalid app>`
			general	

8	Command substitution doesn't replace newlines in the subcommand output with spaces.	`echo `ls``	general	
9	Command substitution and globing doesn't work properly.	`echo `echo *``	general	

Bug #	Description (What's wrong?)	Test Case	Type	Comments (for grading)
10	Project Description specifies that command substitution should be performed before argument splitting. However the group specifies a contradictory assumption (point #2 under Quoting): that spaces in the subcommand output are not used to further split the output into tokens.	N/A	general	
11	Command substitution doesn't allow apps to be specified by the output of the subcommand.	`` `echo ls` ``	general	
12	Command subsitution throws an invalid app exception when empty backticks are used, because the team's implementation of shell interpreted the empty string between the semicolons as an invalid app. This should not happen as the command is still valid although the backticks are redundant.	``ls `` ``	general	
13	Globbing output has redundant "./" in front of each result.	`echo *`	general	
14	Globbing output is not sorted in alphabetical order.	`echo *`	general	

15	Redirection performs the read/write operations on a file in the wrong directory if we perform an operation in a directory other than the project root directory.
	
	Refer to test case for an example. `abc.txt` will not be created in the `testresource` directory and 'c' will not be read from `testresource` directory as well
	
	This fails for any applications that uses redirect.
		`cd testresource;`
		`echo hi > abc.txt`
		`cmp - b < c`
			general	

16	octal value of characters from extended ascii table not translated correctly. e.g. Ç gave a negative octal value	`cmp <file with extended ascii> <file with different content>`	cmp	
17	no warning about EOF e.g. EOF on a.txt after char 8	`cmp -l <fileA> <fileB longer length>`	cmp	
18	-c flag printed decimal value instead of characters	`cmp -c <fileA> <fileB>`	cmp	
19	-ls flag combination should not be acceptable	`cmp -ls <fileA> <fileB>`	cmp	

Bug #	Description (What's wrong?)	Test Case	Type	Comments (for grading)
20	`-cs` flag combination should not be acceptable	`cmp -cs <fileA> <fileB>`	cmp	
21	"b - differ: char 1, line 1" order of b and '-' is flipped	`cmp a b | cmp - b`	cmp	
22	"line 3" and "line 5" should not be a diff since they appear in both files	`diff ./testresource/test_files/diff/A.txt ./testresource/test_files/diff/B.txt`	diff	
23	`diff` two files with trailing newline should not show the newline as a diff	`diff A.txt B.txt`	diff	

24	echo output is incorrectly whitespace-trimmed.	`echo 'a    '`
		or
		`echo '123  ' > file (file should contain trailing whitespaces, but does not)`
			echo	

25	grep on multiple files does not tell you which file the line is from
	expected:
	`grepfile1.txt: line from grepfile1`
	`grepfile2.txt: line fron grepfile 2`
	actual:
	`line from grepfile1`
	`line from grepfile2`
		`grep line team3_resources/grepfile1.txt team3_resources/grepfile2.txt`	grep	

26	grep with flag anywhere except after the grep keyword prints empty result	`grep hello -v ./testresource/test_files/helloworld.txt`	grep	
27	grep with flag anywhere except after the `grep` keyword prints empty result although file is invalid. An error message should be printed to tell the user the file does not exist.	`grep line -v <non-existent file>`	grep	
28	`ls -dR` throws an exception. The team added an assumption that `-d` and `-R` cannot be combined into `-dR`, but we think this assumption is invalid because it changes the specifications.	`ls -dR`	ls	
29	`ls -dd` throws an exception, but it should execute normally as if the input was `ls -d`	`ls -dd`	ls	

Bug #	Description (What's wrong?)	Test Case	Type	Comments (for grading)
30	`ls` output is not sorted in alphabetical order.	`ls`	`ls`	
31	Should throw exception/fail when creating existing directory instead of ignoring	`mkdir bar` `mkdir bar`	`mkdir`	
32	Should fail because intermediate directory does not exist	`mkdir bar/foo`	`mkdir`	
33	Project Description specifies that when no files are specified, use stdin. However the team misinterpreted this to assume that (point #3 under paste): when files are specified, ignore stdin completely. They did not implement mergeFileAndStdin as a result.	`cat ./testresource/sentences1.txt | paste ./testresource/sentences2.txt`	`paste`	
34	`sed` fails all the time if we were to `cd` into another directory.	`cd testresources;` `sed s/paper/REPLACED/ sentences1.txt`	`sed`	
35	Incorrect `sed` commands do not throw exception/show error message - a correct `sed` command should always end with either a number or the delimiter	`sed szhelloz ./testresource/helloworld1.txt`	`sed`	
36	Regex portion of `sed` should not be empty - error message should be shown	`sed s//abc/ ./testresource/helloworld1.txt`	`sed`	
37	Exception should be thrown when sed command is invalid AND no input file is given - this just fails silently now	`sed s//abc`	`sed`	
38	Using * as separator in command leads to the shell treating it as a glob character	`sed s*hello*replaced* ./testresource/helloworld1.txt`	`sed`	
39	file exists but file not found exception thrown. relative directory not handled correctly see this code snippet for more details: https://goo.gl/na9ryG	`split <valid relative file path>`	`split`	

Bug #	Description (What's wrong?)	Test Case	Type	Comments (for grading)
40	should not accept -1 for line number	`split -l -1 <valid file path>`	split	

41	should reject 0 byte size, split when into infinite loop and created infinite files	`split -b 0 <valid file path>`
		`split -b 0b <valid file path>`
		`split -l 0 <valid file path>`
			split	

42	wrong prefix character for more than 676 files. prefix is 'z' in linux distro naming convention is in assumption but specs are already given in the project, so should the assumption be considered an assumption?	`split -l 1 <file with > 676 lines>`	split	

43	no such file or directory thrown, should be error in the flag	`split -bl <valid file path>`
		`split -lb <valid file path>`
		`split -bbbb <valid file path>`
			split	

44	missing split interface, missing `splitStdinByLines`, `splitStdinByBytes`	N/A	split	
45	should split files with 'qqq' prefix but threw '-' no such file or directory	`cat <valid file>	split -l 100 - qqq`	split	
46	file naming sequence is incorrect. xzz jumps to x~aaa. project specs is different from linux distro implementation (xyz xzaaa) naming convention is in assumption but specs are already given in the project, so should the assumption be considered an assumption?	`split -l 1 <large file>`	split	