Rebuttal for team 2 from team 6
2020.4.6

Some Declartion:

We considered all bug related to file permission as invalid.

Bug Report Number	Status	Reason if invalid	Comments (this column should be left empty; for marking usage)
1	invalid	Linux can not handle this case either except with the flag “--”. And we are not required to implement this flag according to the project description.	
2	invalid	Permission related tests are not required.	
3	valid		
4	invalid	Permission related tests are not required.	
5	invalid	Permission related tests are not required.	
6	invalid	The replacement index is counted in each line rather than the whole document. Ref. project specification, page 14, 9.4. sed, in description “For each line containing a match to a specified pattern (in JAVA format), replaces the matched substring with the specified string”	
7	invalid	Permission related, not very clear about what ‘valid’ means.	
8	valid		
9	invalid	duplicate with 8	
10	valid		
11	invalid	There is no specification related that the sub folder is already diff but the content of the sub folder	

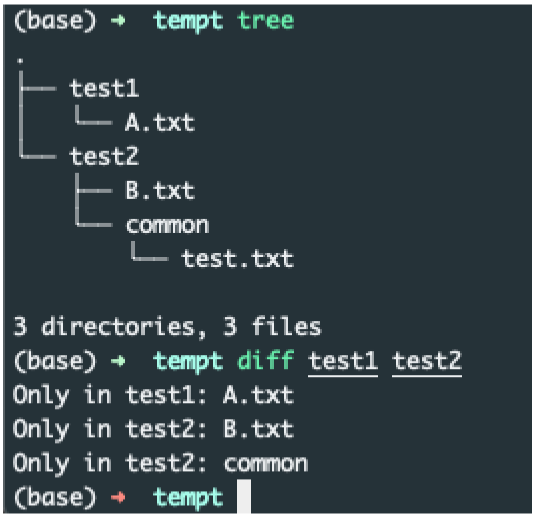
		should be displayed. Refer to the shell behavior No.11.	
12	invalid	Permission related tests are not required.	
13	invalid	Grep hangs the application because the shell is waiting for stdin. Users can exit the shell with ctrl+D and the shell will output the correct stdin loaded before. We will further improve the implementation to let the shell output the stdin once it receives stdin, just as what Linux shell does. Refer to the “Shell behavior for 13” and “Reproduction for 13”.	
14	invalid	This is the correct result. After meeting a back quote, the second double quote is disabled. The command hangs because it is waiting for another back quote and another double quote. The linux shell works in the same way. Refer to the “Shell behavior for 14”	
15	valid	Source file was accidently deleted.	
16	valid	We do throw exceptions but the message is not concrete.	
17	valid	Source file was accidently deleted when moved under a folder.	
18	valid		
19	valid		
20	invalid	Command \`ls\` and \`ls *\` are different. \`ls\` return all the files and folders in the current directory. \`ls *\` return all the files and folders (together with the files and folders in it) in the current directory. While listing files and folders within a folder in the current directory, we need new lines. Our shell works exactly the same as the linux shell. Refer to the “Shell behavior for 20”.	
21	valid		

22	invalid	duplicate with 3	
23	invalid	This test case just proves the correctness of our implementation. Our behavior is the same with a MacOS (Unix) shell. Refer to the “Shell behavior for 23”at the end.	
24	valid		
25	invalid	duplicate with bug 21	
26	invalid	`find` is not meant to be found in every place on the computer, it only finds in the current directory that the user is in. So, `find D:\ -name “test2.txt”` will not work because it can’t find the folder named “D:\” in the current directory.	
27	invalid	`sort -r` is not meant to reverse the output, but to sort the output in the opposite order of `sort`. Since the output of `paste test1.txt` is already in the right order (from large to small according to ASCII value), `sort -r` will change anything. Ref. project specification, page 19, 9.13. sort, in description: “Within each character class, the characters are sorted according to their ASCII value.” in Command format: “-r - Sort in reverse order. ”	
28	valid	The table in the assumption is not up to date	
29	invalid	We implemented the paste according to the example here https://www.geeksforgeeks.org/paste-command-in-linux-with-examples/ 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.	
30	invalid	The number is too large. UNIX-like (MacOS and Ubuntu 18.04) shells also consider this command as invalid. Refer to “Shell behavior for 30”	

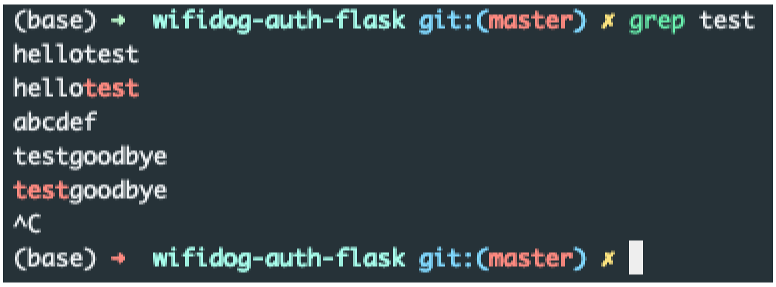
Shell behavior

Shell behaviour for 11: (Mac OS)



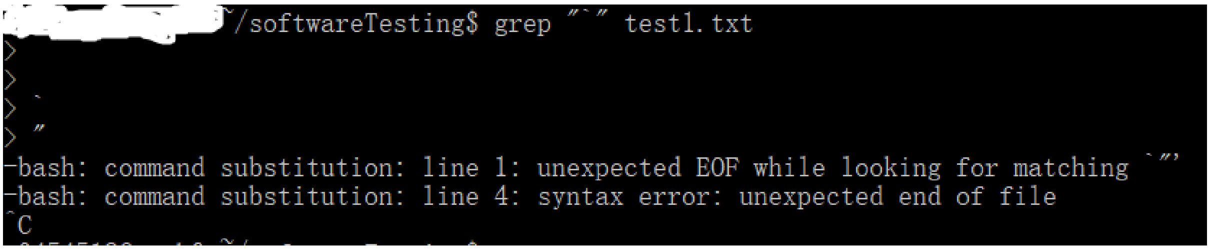
```
(base) ➜  tempt tree
.
├── test1
│   └── A.txt
└── test2
    ├── B.txt
    └── common
        └── test.txt

3 directories, 3 files
(base) ➜  tempt diff test1 test2
Only in test1: A.txt
Only in test2: B.txt
Only in test2: common
(base) ➜  tempt
```

Shell behaviour for 13: (Mac OS)



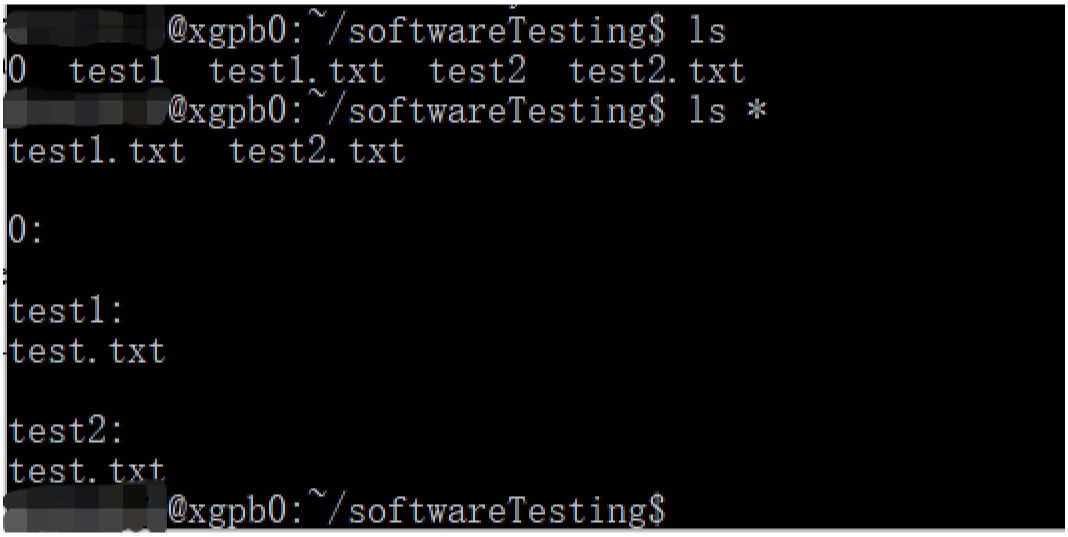
```
(base) ➜  wifidog-auth-flask git:(master) ✗ grep test
hellotest
hellotest
abcdef
testgoodbye
testgoodbye
^C
(base) ➜  wifidog-auth-flask git:(master) ✗
```

Shell behaviour for 14: (Linux)



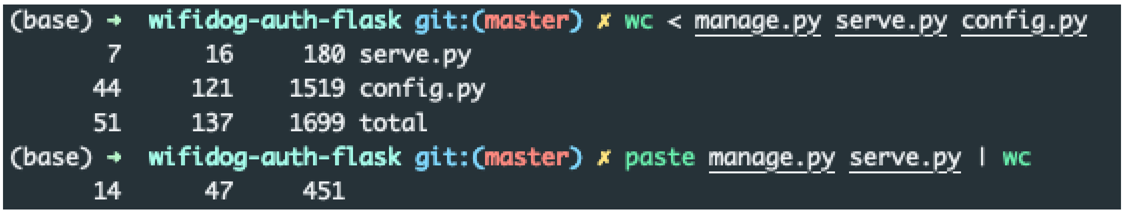
```
[redacted]~/softwareTesting$ grep "`" test1.txt
>
>
> `
> "
-bash: command substitution: line 1: unexpected EOF while looking for matching `"'
-bash: command substitution: line 4: syntax error: unexpected end of file
^C
```

Shell behaviour for 20:(Linux)



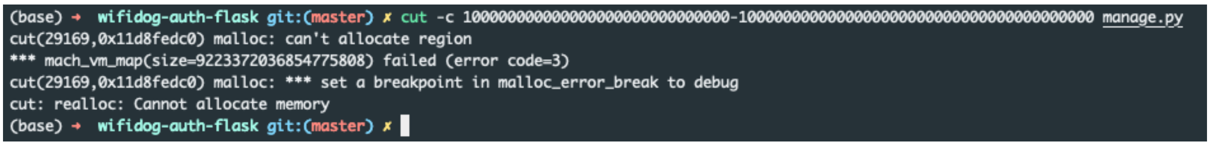
```
[redacted]@xgpb0:~/softwareTesting$ ls
0  test1  test1.txt  test2  test2.txt
[redacted]@xgpb0:~/softwareTesting$ ls *
test1.txt  test2.txt

0:

test1:
test.txt

test2:
test.txt
[redacted]@xgpb0:~/softwareTesting$
```

Shell behavior for 23: (MacOS)



```
(base) → wifidog-auth-flask git:(master) ✗ wc < manage.py serve.py config.py
       7      16     180 serve.py
      44     121    1519 config.py
      51     137    1699 total
(base) → wifidog-auth-flask git:(master) ✗ paste manage.py serve.py | wc
      14      47     451
```

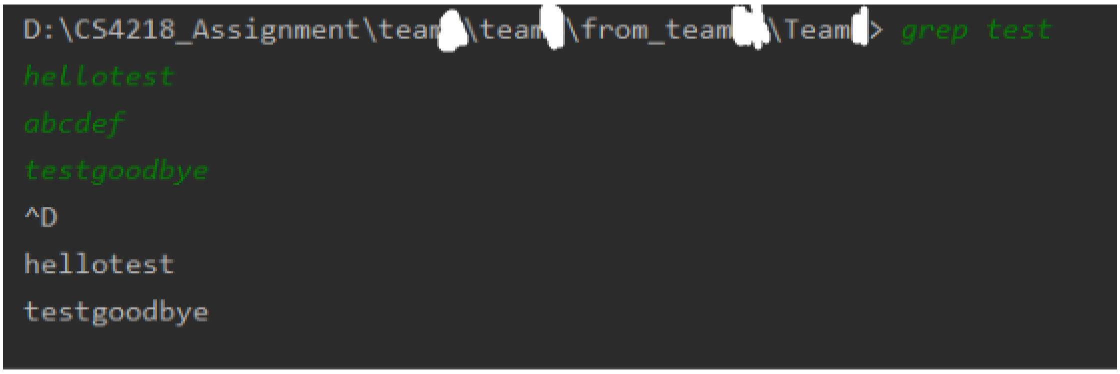
Shell behavior for 30: (MacOS)



```
(base) → wifidog-auth-flask git:(master) ✗ cut -c 100000000000000000000000000000-1000000000000000000000000000000000000000 manage.py
cut(29169,0x11d8fedc0) malloc: can't allocate region
*** mach_vm_map(size=9223372036854775808) failed (error code=3)
cut(29169,0x11d8fedc0) malloc: *** set a breakpoint in malloc_error_break to debug
cut: realloc: Cannot allocate memory
(base) → wifidog-auth-flask git:(master) ✗ 
```

Reproductions:

Reproduction for 13



```
D:\CS4218_Assignment\team[redacted]\team[redacted]\from_team[redacted]\Team[redacted]> grep test
hellotest
abcdef
testgoodbye
^D
hellotest
testgoodbye
```