

PRINT Name: _____ LAB Section: **One-Answer Multiple Choice 234 Questions****Weight 15%**

- ☞ Read **all** the words of these instructions and **both** sides (back and front) of all pages.
- ☞ Manage your time. Answer questions you know, first. One Answer per question.
- ☞ **PRINT** your Name and Lab on this Question Sheet. You may write or draw on this sheet.
- ☞ Use your full, unabbreviated name on the mark-sense form. Do not abbreviate your name.
- ☞ Enter your NAME, Student Number, and Answers. Fill in the bubbles with pencil, no pen.
- ☞ The answer to the questions below about reading/doing all these test instructions is: **Jes**

191. Answer **191** is **E**
192. Answer **192** is **E**
193. Answer **193** is **C**
194. Answer **194** is **A**
195. Answer **195** is **B**
196. Answer **196** is **D**

Your Test Version is:

E E C A B D

Fill in the bubbles for the above six letters as six answers **191** through **196** on the back side of the Scantron form, in the lower-right-most answer column.

1. **Did you read all the words of the test instructions on page one?**
 - a. **Taip** (Yes - Lithuanian)
 - b. **Sim** (Yes - Portuguese)
 - c. **Igen** (Yes - Hungarian)
 - d. **Tak** (Yes - Polish)
 - e. **Jes** (Yes - Esperanto)
2. **My three-digit Lab Section number is:**
 - a. The timetable section number of my weekly 2-hour lab period.
 - b. My lecture room number, e.g. **T130**, **T117**
 - c. The Test Version number printed in the top left corner.
 - d. My lab room number, e.g. **B384**, **T321**, **J218**, **N201**
 - e. My lecture section number, e.g. **010** or **020**.
3. What is the link count of directory **d** after these successful commands?
`mkdir d ; touch f ; cd d ; ln ../f x`
 - a. 3
 - b. 5
 - c. 1
 - d. 2
 - e. 4
4. What is the link count of directory **d** after these successful commands?
`mkdir d ; mkdir d/a ; mkdir d/a/b ; mkdir d/a/c`
 - a. 2
 - b. 5
 - c. 1
 - d. 4
 - e. 3
5. What is the link count of file **f** after these successful commands?
`rm f ; touch f ; ln f bar`
`cp bar x ; ln x y ; ln bar z ; ln z a`
 - a. 2
 - b. 5
 - c. 1
 - d. 4
 - e. 3
6. What is the link count of directory **d** after these successful commands?
`mkdir d d/a d/b d/c d/c/z ; touch d/x d/y`
 - a. 5
 - b. 6
 - c. 2
 - d. 4
 - e. 3

7. If the file **bat** contained the word **foo**, what is the output on your screen after this: `PATH=/bin/cat:/bin/who:/bin/ls ; cat bat`
 - a. **bat**
 - b. **bash: cat: command not found**
 - c. **foo**
 - d. no output on screen
 - e. **cat: bat: No such file or directory**
8. What is the output on your screen after these command lines:
`echo one >x ; ln x y ; echo ten >y`
`echo two >x ; cat y`
 - a. **one** followed by **ten** and **two**
 - b. no output on screen
 - c. **two**
 - d. **one**
 - e. **ten**
9. What is usually in the environment variable **\$HOME**?
 - a. the relative path of the system **/home** directory
 - b. the relative path of the **ROOT** directory
 - c. the absolute path of your login home directory
 - d. the relative path of your login home directory
 - e. the absolute path of the system **/home** directory
10. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:

```
111 -rw-r--r-- 1 me me 100 Jan 1 1:00 a
222 -rw-r--r-- 2 me me 100 Jan 1 1:00 b
333 -rw-r--r-- 2 me me 100 Jan 1 1:00 c
444 -rw-r--r-- 1 me me 100 Jan 1 1:00 d
```

 - a. 3
 - b. 4
 - c. 0
 - d. 1
 - e. 2
11. In an empty directory, how many words are in file **c** after this:
`touch a ; mv b a >b ; ls >c`
 - a. 3
 - b. 1
 - c. 0
 - d. 2
 - e. 4
12. How many arguments are passed to the command by the shell:
`<foo foo " a 'b c' d " e f ' g " h " ' >foo`
 - a. 6
 - b. 4
 - c. 5
 - d. 3
 - e. 2
13. How many arguments are passed to the command by the shell:
`<pig pig -x " " -z -r" " >pig pig pig`
 - a. 7
 - b. 5
 - c. 6
 - d. 9
 - e. 8
14. Which command finds your account login userid in the password file?
 - a. `find /etc/passwd -name $USER`
 - b. `find $USER /etc/passwd`
 - c. `cat $USER /etc/passwd`
 - d. `fgrep $USER /etc/passwd`
 - e. `fgrep /etc/passwd $USER`

15. Create a symbolic link under `/usr` named `bar` that has target `xy`:
- `ln -s 'xy' '/usr/bar'`
 - `ln -s /usr/bar '/usr/xy'`
 - `ln -s '/usr/xy' /usr/bar`
 - `ln -s /usr/bar 'xy'`
 - `ln -s 'xy' /bar/usr`
16. How many files are touched? `touch 1 "2 3" ' ' 4 5`
- 5
 - 4
 - 7
 - 3
 - 6
17. Which command line allows programs in the current directory to execute without preceding the names with `./`? (P.S. Security Risk! Don't do this!)
- `PATH=/usr/bin/./:$HOME`
 - `$PATH=.:$HOME:/usr/bin`
 - `PATH=/bin:/usr/bin:.`
 - `$PATH=/usr/bin:./bin`
 - `PATH=./$HOME:/usr/bin`
18. You enter this `cp a/b c/` and get `cp: a: No such file or directory` because:
- pathname `a` exists but is a file, not a directory
 - the command `cp` is not in your search `PATH`
 - directory `a` does not exist
 - you forgot to specify the destination file name after `c/`
 - directory `c` does not exist
19. If you are in `/bin` and `ls -l` shows a symbolic link `foo -> /bar` then dereference the absolute path of `foo` with no symbolic links:
- `/bar`
 - `/bin/bar/foo`
 - `/bin/foo/bar`
 - `/foo/bar`
 - `/bin/bar`
20. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
- ```
111 -rw-r--r-- 1 me me 1 Jan 1 1:00 a
222 -rw-r--r-- 1 me me 1 Jan 1 1:00 b
333 -rw-r--r-- 1 me me 1 Jan 1 1:00 c
444 -rw-r--r-- 2 me me 1 Jan 1 1:00 d
```
- 3
  - 0
  - 4
  - 1
  - 2
21. If file `foo` occupies one disk block, how many disk blocks are in use after this:
- ```
cp foo bar ; ln bar one ; cp one two ; ln one pig
```
- 2
 - 5
 - 1
 - 3
 - 4
22. If I am in directory `/tmp` and `mt` is an empty sub-directory, what is true after this:
- ```
touch mt/bar ; mkdir mt/me ; cp mt/bar mt/./me
```
- there is a second copy of the file `bar` in directory `mt`
  - the directory `mt` now contains only a file named `me`
  - the directory `mt` is now empty
  - the command fails because the name `mt/./me` does not exist
  - there is a second copy of the file `bar` in file `/tmp/me`

23. What is the output on your screen after this: `echo hi >out | wc -w`
- no output
  - 1
  - 2
  - 0
  - 3
24. What is the output of this in an empty directory:
- ```
date >.date ; users >.users ; echo .?*
```
- an error message from `echo` saying `.*` does not exist
 - `.date`
 - `.. .date .users`
 - `.*`
 - `.date .users`
25. Which command line has exactly one argument?
- `echo "It's "'funny how'" it's done."`
 - `echo "It's "'funny how' " it's done."`
 - `echo 'It's "funny how" it's done.'`
 - `echo 'It's "'funny how'" it's done.'`
 - `echo "It's " 'funny how'" it's done."`
26. The option to `ls` that shows inode (index) numbers is:
- `-x`
 - `-i`
 - `-a`
 - `-l`
 - `-1`
27. Which command moves a file into the parent directory?
- `mv file ..`
 - `mv file,..`
 - `mv .. file`
 - `mv ../file`
 - `mv file/..`
28. Which command removes *only* this four-character name containing a special character: `?xyz`
- `rm '?xyz'`
 - `rm '?xyz'`
 - `rm ''?xyz''`
 - `rm '?xyz'`
 - `rm ?xyz`
29. In an empty directory, what is the output on your screen after this:
- ```
echo one >.bar ; ls .????*
```
- an error message from `ls` saying `????*` does not exist
  - `.. .bar`
  - `????*`
  - `one`
  - `.bar`
30. If `/bin/foo` is a program that outputs `one` and `/usr/bin/foo` is a program that outputs `two`, what is the output on your screen after this:
- ```
PATH=/etc:/usr/bin:/usr:/bin:/dev ; foo
```
- `two`
 - `bash: foo: command not found`
 - `two` followed by `one`
 - `one`
 - `one` followed by `two`

31. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
- ```
111 -rw-r--r-- 2 me me 100 Jan 1 1:00 a
111 -rw-r--r-- 2 me me 100 Jan 1 1:00 b
222 -rw-r--r-- 2 me me 100 Jan 1 1:00 c
222 -rw-r--r-- 2 me me 100 Jan 1 1:00 d
```
- a. 4            b. 0            c. 1            d. 2            e. 3
32. What is true about this output from `ls -il foo bar`
- ```
15 -r-x----- 2 me me 3 Jan 1 1:00 foo
15 -rwxrwxrwx 2 me me 3 Jan 1 1:00 bar
```
- a. this output is not possible
b. `foo` and `bar` are two of three names for the same file
c. `foo` and `bar` each have three names (six names total)
d. `foo` and `bar` are names for different files
e. `foo` and `bar` are names for the same file
33. What is in file `c` after this:
- ```
echo foo >a ; ln a b ; echo bar >>b ; ln a c ; rm a
```
- a. `bar`                                            b. `foo`  
c. no such file (nonexistent)                    d. `foo` followed by `bar`  
e. nothing (empty file)
34. What is the link count of file `f` after these successful commands?
- ```
rm f ; touch f ; cp f x
ln f a ; ln x y ; ln a z ; ln z q
```
- a. 5 b. 4 c. 3 d. 6 e. 2
35. In an empty directory, what is the output on your screen after this:
- ```
touch A a ; echo * >"*" ; ls
```
- a. `A a >A a`                                      b. `A a >*`                                      c. No output  
d. `* A a`                                        e. `* >*`
36. In an empty directory, what is the output on your screen after this:
- ```
touch 1 2 3 ; cow="*" ; echo ""$cow"
```
- a. `"1 2 3"` b. `*` c. `"$cow"`
d. `$cow` e. `1 2 3`
37. How many files are touched? `touch 1 "2 3" ' 4 ' 5`
- a. 6 b. 4 c. 7 d. 5 e. 3
38. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
- ```
111 -rw-r--r-- 1 me me 100 Jan 1 1:00 a
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 b
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 c
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 d
```
- a. 4            b. 1            c. 2            d. 0            e. 3

39. What is the link count of directory `dir` after these successful commands?
- ```
mkdir dir ; cd dir ; touch a b c ; mkdir d e
```
- a. 3 b. 4 c. 5 d. 7 e. 2
40. Which command removes *only* this five-character name containing a special character: `date`?
- a. `rm date*` b. `rm date\\?` c. `rm ./date?`
d. `rm ./date\?` e. `rm date/?`
41. What is the output on your screen of this unquoted command line:
- ```
mkdir a ; touch b a/b1 a/b2 ; find a -name b*
```
- a. `a/b1 a/b2`                                    b. `b a/b1 a/b2`                                    c. no output  
d. `b`                                            e. `b1 b2`
42. What is the link count of directory `d` after these successful commands?
- ```
mkdir d ; mkdir d/a ; mkdir d/b ; mkdir d/b/c
```
- a. 1 b. 2 c. 5 d. 3 e. 4
43. Which command recursively finds all things named `foo`?
- a. `cat -name foo` b. `echo -name foo`
c. `find -name foo` d. `fgrep -name foo`
e. `ls -name foo`
44. If your `PATH` contained only the file names `/bin/sh`, `/bin/cat`, and `/bin/ls`, then what is the output on your screen of this command:
- ```
cat /etc/passwd
```
- a. `cat: bash: no such file or directory`  
b. `cat: /etc/passwd: command not found`  
c. `bash: /bin/sh: command not found`  
d. `bash: /bin/cat: no such file or directory`  
e. `bash: cat: command not found`
45. What is the usual output on your screen of this:
- ```
mkdir dir ; cd dir >foo ; cat foo
```
- a. `cat: foo: No such file or directory`
b. no output
c. `dir`
d. `foo`
e. `bash: cd: dir: No such file or directory`
46. How many arguments are passed to the command by the shell:
- ```
<wc wc " 1 '2 3' 4 " 5 6 ' 7 " 8 " ' >wc 9
```
- a. 3            b. 5            c. 6            d. 2            e. 4
47. How many files are touched? `touch 1 "2 3 ' 4 '" 5`
- a. 2            b. 1            c. 5            d. 4            e. 3
48. What is the link count of directory `foo` after these successful commands?
- ```
mkdir foo ; cd foo ; touch a b c
```
- a. 5 b. 1 c. 2 d. 4 e. 3

49. What is in the local variable `$$` ?
- the process ID of the current shell
 - the cpu cost of the current session, in dollars
 - the command name of the previous command line
 - `$$` is not a valid variable name
 - the first argument of the previous command line
50. What is the link count of file `foo` after these successful commands?
- ```
rm foo ; touch foo ; ln foo bar
cp bar x ; ln x y ; ln bar z ; ln z a
```
- 2
  - 5
  - 3
  - 4
  - 1
51. Which command line always prints just the two characters `$x` on the screen?
- `echo $x`
  - `echo "$$x"`
  - `echo $$x`
  - `echo '$x'`
  - `echo "$x"`
52. In an empty directory, what is the output on your screen after this:
- ```
touch a ; ls | wc -w
```
- no output
 - 1
 - 2
 - 0
 - 3
53. The correct syntax to assign to a shell variable is:
- `V="foo bar"`
 - `V=foo bar"`
 - `V = "foo bar"`
 - `V = foo bar`
 - `V=foo bar`
54. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are output on your screen by this: `cat b | cat a`
- 5
 - 3
 - 2
 - 2 followed by 3
 - 3 followed by 2
55. How many arguments are passed to the command by the shell:
- ```
<cow cow "-x" -y '-z' >cow cow
```
- 5
  - 3
  - 6
  - 4
  - 2
56. Which of the following is true, given this long directory listing:
- ```
drwxr-x--x 128 me me 32 Jan 1 1:00 dir
```
- The number 128 is the inode number of this directory.
 - The number 32 is the count of links (names) this directory has.
 - The number 128 is the count of links (names) this directory has.
 - The number 32 is the inode number of this directory.
 - The number 128 is the size of this directory.

57. What is true about this output from `ls -il foo bar`
- ```
35 -rw-rw-r-- 2 me me 3 Jan 1 1:00 foo
36 -rw-rw-r-- 2 me me 3 Jan 1 1:00 bar
```
- this output is not possible
  - `foo` and `bar` each have two names (four names total)
  - `foo` and `bar` are two of three names for this file
  - `foo` and `bar` each have three names (six names total)
  - `foo` and `bar` are names for the same file
58. Given this `ls -il` long listing:
- ```
123 drwxr-xr-x 456 me me 789 Jan 1 1:00 dir
```
- How many subdirectories lie immediately under `dir`?
- 454
 - 787
 - 123
 - 456
 - 789
59. If `/bin/prg` is a program that outputs `hi` and `/usr/bin/prg` is a program that outputs `foo` what is the output on your screen after this:
- ```
PATH=/etc:/usr/bin:/bin ; prg
```
- `foo`
  - `foo` followed by `hi`
  - `bash: prg: command not found`
  - `hi` followed by `foo`
  - `hi`
60. If file `foo` occupies one disk block, how many disk blocks are in use after this:
- ```
cp foo bar ; ln bar one ; cp one two ; cp one xxx
```
- 3
 - 4
 - 2
 - 5
 - 1
61. Which command line would show the index (inode) number of a file?
- `ls -i file`
 - `ls -l file`
 - `cat -i file`
 - `cat -l file`
 - `find -i file`
62. What is usually in the environment variable `$PATH`?
- a colon-separated list of directories containing command names
 - a colon-separated list of your `passwd` file fields
 - the absolute path of the system `/path` directory
 - the absolute path of your login shell
 - the absolute path of your login home directory
63. What is the link count of file `f` after these successful commands?
- ```
rm f ; touch f ; ln f bar
cp bar x ; ln x y ; ln bar z
```
- 3
  - 5
  - 1
  - 2
  - 4
64. What is the link count of file `foo` after these successful commands?
- ```
rm foo ; touch foo ; ln foo bar
cp bar a ; ln a b ; ln bar c ; cp c a
```
- 4
 - 3
 - 5
 - 2
 - 1

65. Which command line outputs inode/filename pairs for names in the current directory, sorted by inode number?
- a. `sort -n | ls -ai` b. `ls -i * > sort -n`
 c. `ls /* | sort -node` d. `ls -node * > sort -n`
 e. `ls -ai | sort -n`
66. If `/bin/xxx` is a program that outputs `one` and `/usr/bin/xxx` is a program that outputs `two`, what is the output on your screen after this:
- ```
PATH=/usr:/usr/bin:/etc:/bin ; xxx
```
- a. `two`  
 b. `two` followed by `one`  
 c. `one` followed by `two`  
 d. `bash: xxx: command not found`  
 e. `one`
67. To change to the parent directory, do this:
- a. `pwd`                                      b. `cd ..`                                      c. `cd .`  
 d. `cd`                                        e. `pwd ..`
68. In an empty directory, what is the output on your screen after this:
- ```
echo one >.bar ; echo .*
```
- a. `one`
 b. `.bar`
 c. `.. .bar`
 d. an error message from `echo` saying `.*` does not exist
 e. `.*`
69. What is in file `foo` after this:
- ```
echo hi >a ; ln a b ; echo me >b ; ln a foo ; rm a b
```
- a. `me`                                        b. no such file (nonexistent)  
 c. nothing (empty file)                    d. `hi`  
 e. `hi` followed by `me`
70. What is the output on your screen after this:
- ```
echo 1 >x ; ln x y ; echo 2 >>y ; sort x
```
- a. 2 followed by 1 b. no output c. 1
 d. 1 followed by 2 e. 2
71. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `e` after this: `ln a d ; cp a f ; ln d c ; ln c e ; cat a b d f >e`
- a. 6 b. 9 c. 5 d. 2 e. 3
72. What command will recursively show disk usage in directories?
- a. `find` b. `tree` c. `df` d. `ls` e. `du`
73. How many files are touched? `touch '1' '2 3' '4'" ' 5`
- a. 2 b. 3 c. 5 d. 1 e. 4

74. What is in file `c` after this:
- ```
echo A >a ; ln a b ; echo B >b ; ln a c ; rm a b
```
- a. no such file (nonexistent)                b. `A`  
 c. `A` followed by `B`                        d. `B`  
 e. nothing (empty file)
75. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
- ```
111 -rw-r--r-- 1 me me 100 Jan 1 1:00 a
222 -rw-r--r-- 1 me me 100 Jan 1 1:00 b
333 -rw-r--r-- 1 me me 100 Jan 1 1:00 c
444 -rw-r--r-- 1 me me 100 Jan 1 1:00 d
```
- a. 1 b. 4 c. 3 d. 0 e. 2
76. What does *quoting* mean on a shell command line?
- a. turning off the special meaning of shell meta-characters
 b. typing a "control" character using the [CTRL] key
 c. using a leading tilde ("~") on a pathname to mean your **HOME** directory
 d. setting the `PS1` variable to be your shell prompt
 e. using more than one pathname argument to a command, e.g. `rm a b c`
77. What is usually in the environment variable `$SHELL`?
- a. the absolute path of your login shell
 b. the relative path of the system `/shell` directory
 c. the relative path of the `/home/shell` directory
 d. the relative path of your login shell
 e. the absolute path of the system `/shell` directory
78. What is the output on your screen after these command lines:
- ```
echo 1 >x ; ln x y ; echo 2 >>y
head -1 x >y ; cat y
```
- a. 1                                            b. 2                                            c. 2 followed by 1  
 d. 1 followed by 2                            e. no output
79. Dereference the following symlink `xyz` into its equivalent absolute path:
- ```
ln -s ../../a/./b/./bar /tmp/a/b/xyz
```
- a. `/tmp/a/bar` b. `/tmp/b/bar` c. `/tmp/b/xyz`
 d. `/tmp/a/b/bar` e. `/tmp/bar`
80. If the file `bat` contained the word `foo`, what is the output on your screen after this: `PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/ls bat`
- a. no output on screen
 b. `bash: /bin/ls: command not found`
 c. `foo`
 d. `bat`
 e. `/bin/ls: bat: No such file or directory`

81. How many arguments are passed to the command by the shell:
`echo " 1 2 " three ' 4 ' five"6"`
 a. 5 b. 4 c. 3 d. 9 e. 1
82. Which command recursively finds all things with names beginning with `foo`?
 a. `find -name 'foo*'` b. `ls -name foo*`
 c. `fgrep -name foo?` d. `ls foo*`
 e. `find -name foo?`
83. How many arguments are passed to the command by the shell:
`echo "cow "y " bat 'man x' " pig'a "hop' a b`
 a. 11 b. 6 c. 7 d. 4 e. 5
84. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
`111 -rw-r--r-- 2 me me 100 Jan 1 1:00 a`
`222 -rw-r--r-- 1 me me 100 Jan 1 1:00 b`
`333 -rw-r--r-- 2 me me 100 Jan 1 1:00 c`
`333 -rw-r--r-- 2 me me 100 Jan 1 1:00 d`
 a. 4 b. 1 c. 2 d. 0 e. 3
85. Rewrite as a simplified absolute path:
`/home/me/../../you/../../../../etc/../../home/me/../../you/../../me/../../foo`
 a. `/home/foo` b. `/home/you/foo`
 c. `/etc/foo` d. `/foo`
 e. `/home/me/foo`
86. How many arguments are passed to the command by the shell:
`echo 'It's "1 2" isn't it? I can't decide.`
 a. 3 b. 4 c. 5 d. 2 e. 6
87. How many arguments are passed to the command by the shell:
`<foo foo " a 'b c' d " e ' f " g " ' >foo`
 a. 5 b. 6 c. 2 d. 4 e. 3
88. What is the link count of directory `z` after these successful commands?
`mkdir z ; cd z ; touch a b ; mkdir c d e`
 a. 6 b. 7 c. 4 d. 5 e. 3
89. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
`111 -rw-r--r-- 1 me me 100 Jan 1 1:00 a`
`222 -rw-r--r-- 1 me me 100 Jan 1 1:00 b`
`444 -rw-r--r-- 2 me me 100 Jan 1 1:00 c`
`444 -rw-r--r-- 2 me me 100 Jan 1 1:00 d`
 a. 2 b. 0 c. 1 d. 4 e. 3
90. If your terminal type is `xterm`, what is the output of this: `echo '$TERM'`
 a. `xterm` b. `$TERM`
 c. `'xterm'` d. `'$TERM'`
 e. no output on screen

91. If I have a directory named `a/b`, which action would increase its *link count* by exactly one?
 a. create a directory named `a/b/c`
 b. create a directory named `a/b2`
 c. create a file named `a/b2`
 d. create a hard link to directory `b` named `b2`
 e. create a file named `a/b/c`
92. What is the output on your screen after this:
`echo one >x ; ln x y ; echo two >>y ; sort x`
 a. `one` b. no output
 c. `two` followed by `one` d. `one` followed by `two`
 e. `two`
93. In an empty directory, what is the output on your screen after this:
`echo hi >a ; ls | wc -w`
 a. 2 b. 1 c. a
 d. 0 e. no output
94. In an empty directory, what is the output on your screen after this:
`touch 1 2 3 ; cow="*" ; echo "$cow"`
 a. 1 2 3 b. `$cow` c. *
 d. `"$cow"` e. `"1 2 3"`
95. What is the output on your screen after this:
`echo hi >a ; cp a b | wc -w`
 a. 3 b. no output c. 2
 d. 1 e. 0
96. What is the link count of file `foo` after these successful commands?
`rm foo ; touch foo ; ln foo bar`
`cp bar x ; ln x y ; ln y z`
 a. 1 b. 4 c. 0 d. 2 e. 3
97. Which command copies a directory:
 a. `mv -rf dir1 dir2` b. `cp -r dir1 dir2`
 c. `cp dir1 dir2` d. `mv -r dir1 dir2`
 e. `mv -f dir1 dir2`
98. In an empty directory, what is the output on your screen after this:
`echo hi >a ; mv a b ; ln b c ; ls >wc -l`
 a. a b. no output c. 1
 d. 2 e. 0
99. How many arguments are passed to the command by the shell:
`echo " one '2 three' 4 "five 6 ' 7 "8 ' >out`
 a. 4 b. 2 c. 6 d. 3 e. 5

100. What is the link count of file **f** after these successful commands?
`rm f ; touch f ; cp f x`
`ln f a ; ln x y ; ln a z ; ln x b`
 a. 3 b. 4 c. 6 d. 5 e. 2
101. In an empty directory, how many words are in file **a** after this:
`echo It's redirected >b isn't it\? ; ls >a`
 a. 2 b. 4 c. 1 d. 0 e. 3
102. If file **one** occupies one disk block, how many disk blocks are in use after this:
`cp one foo ; ln foo two ; ln two bar ; ln one cow`
 a. 3 b. 1 c. 5 d. 4 e. 2
103. If **/bin/bat** is a program that outputs **foo** and **/usr/bin/bat** is a program that outputs **bar** what is the output on your screen after this:
`PATH=/usr:/usr/bin:/bin ; bat`
 a. **bar**
 b. **foo**
 c. **bar** followed by **foo**
 d. **foo** followed by **bar**
 e. **bash: bat: command not found**
104. How many arguments are passed to the command by the shell:
`echo ' one two ' three ' four ' 5'6'`
 a. 1 b. 9 c. 6 d. 4 e. 5
105. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:

```
111 -rw-r--r-- 3 me me 100 Jan 1 1:00 a
111 -rw-r--r-- 3 me me 100 Jan 1 1:00 b
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 c
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 d
```

 a. 3 b. 1 c. 2 d. 4 e. 0
106. What is the link count of an empty directory?
 a. 2 b. 4 c. 1 d. 3 e. 0
107. If **/bin/foo** is a program that outputs **mom** and **/usr/bin/foo** is a program that outputs **dad**, what is the output on your screen after this:
`PATH=/dev:/usr/bin:/usr:/bin:/etc ; /bin/foo`
 a. **dad** followed by **mom**
 b. **dad**
 c. **mom**
 d. **mom** followed by **dad**
 e. **bash: /bin/foo: command not found**
108. How many arguments are passed to the command by the shell:
`<foo foo -x " " -z -r" " >foo 'foo foo'`
 a. 8 b. 5 c. 6 d. 7 e. 9

109. What is the link count of directory **d** after these successful commands?
`mkdir d ; mkdir d/a d/b ; touch d/c d/e`
 a. 1 b. 5 c. 4 d. 3 e. 2
110. Which option to **ls** displays the directory itself and not its contents?
 a. **-l** b. **-d** c. **-i** d. **-R** e. **-a**
111. If you are in **/etc** and **ls -l** shows a symbolic link **bar** **-> foo** then dereference the absolute path of **bar** with no symbolic links:
 a. **/bar/foo** b. **/etc/bar/foo** c. **/foo**
 d. **/etc/foo** e. **/etc/foo/bar**
112. How many files are touched? `touch '1 '2 3 '4' '5`
 a. 2 b. 1 c. 3 d. 4 e. 5
113. A "dangling symlink" is a symlink to:
 a. a directory b. a parent directory
 c. a non-existent target d. the current directory
 e. a special device file
114. Which command shows the name of the current computer:
 a. **find** b. **hostname** c. **history**
 d. **comname** e. **whois**
115. In an empty directory, what is the output on your screen after this:
`touch A a ; echo * ">*"`
 a. No output b. **A a >*** c. **A a**
 d. *** >*** e. **A a >A a**
116. If **/bin/foo** is a program that outputs **one** and **/usr/bin/foo** is a program that outputs **two**, what is the output on your screen after this:
`PATH=/bin/ls:/home:/usr/bin/cat:/etc ; foo`
 a. **two** followed by **one**
 b. **one** followed by **two**
 c. **two**
 d. **bash: foo: command not found**
 e. **one**
117. In an empty directory, how many words are in file **c** after this:
`touch a b 1 b a ; ls >c`
 a. 2 b. 4 c. 3 d. 0 e. 1
118. What is the link count of directory **z** after these successful commands?
`mkdir z ; mkdir z/a ; touch z/b z/c z/d`
 a. 5 b. 4 c. 1 d. 3 e. 2

119. What command will recursively find all pathnames named **foo** in **/bin**?
- `find /bin -name 'foo'`
 - `grep /bin -basename 'foo'`
 - `ls -R 'foo' /bin`
 - `grep 'foo' /bin`
 - `find foo -name '/bin'`
120. How many arguments are passed to the command by the shell:
`echo 'It's a bird! It's a plane!'`
- 5
 - 1
 - 3
 - 4
 - 2
121. In an empty directory, what is the output on your screen after this:
`echo one >.bar ; echo .??*`
- `.. .bar`
 - an error message from `echo` saying `.??*` does not exist
 - `.bar`
 - `.??*`
 - `one`
122. What is the link count of directory **dir** after these successful commands?
`mkdir dir ; touch foo ; cd dir ; ln ../foo bar`
- 1
 - 2
 - 3
 - 5
 - 4
123. Which of these statements is true?
- Only backslashes are strong enough to stop GLOB patterns from expanding.
 - If `/x` is an empty directory, `sort /x/*` produces an error message.
 - If `/y` is an empty directory, `echo /y/*` produces an error message.
 - Only double quotes are strong enough to stop GLOB patterns from expanding.
 - Only single quotes are strong enough to stop GLOB patterns from expanding.
124. What is the output on your screen after this:
`mkdir foo ; rmdir foo | wc -w`
- 1
 - 0
 - 3
 - 2
 - no output
125. If I have a directory named **/1/2**, which action would increase its *link count* by exactly one?
- create a directory named `/1/2`
 - create one file named `/1/2/3`
 - create a directory named `/1/2/3`
 - create one file named `/1/22`
 - create a directory named `/1/22`
126. What is the link count of file **f** after these successful commands?
`rm f ; touch f ; ln f bar`
`cp bar x ; ln x y ; ln y z`
- 1
 - 0
 - 3
 - 2
 - 4

127. How many arguments are passed to the command by the shell:
`<cow cow "-x "-y '-z' >cow cow`
- 3
 - 4
 - 5
 - 6
 - 7
128. What is the link count of directory **d** after these successful commands?
`mkdir d ; cd d ; touch f ; ln f a ; ln f b`
- 2
 - 1
 - 5
 - 4
 - 3
129. If **/bin/pig** is a program that outputs **xx** and **/usr/bin/pig** is a program that outputs **foo** what is the output on your screen after this:
`PATH=/home:/bin:/dev:/usr/bin ; pig`
- xx** followed by **foo**
 - xx**
 - foo** followed by **xx**
 - foo**
 - bash: pig: command not found**
130. How many arguments are passed to the command by the shell:
`echo 'And it's not hard, it's just logical.'`
- 7
 - 5
 - 4
 - 3
 - 6
131. Which command line makes a file executable?
- `chmod u+x file`
 - `umask u+x file`
 - `umask -x file`
 - `chmod -x file`
 - `umask u=x file`
132. What is the link count of file **foo** after these successful commands?
`rm foo ; touch foo ; ln foo bar ; ln bar x`
`cp bar a ; ln a b ; ln x c ; cp c d`
- 2
 - 1
 - 5
 - 3
 - 4
133. Which one of these names is usually a shell environment variable?
- FOOBAR**
 - foobar**
 - fooBar**
 - FooBar**
 - FooBar**
134. How many files are touched? `touch "1" 2 3 " " 4 5`
- 4
 - 7
 - 3
 - 6
 - 5
135. Which of the following **PATH** statements makes the most sense?
- `PATH=/bin:/usr/bin:/etc/passwd`
 - `PATH=/bin/bash:/usr/bin:/bin`
 - `PATH=/bin:/etc/passwd:/usr/bin`
 - `PATH=/bin:/usr/bin`
 - `PATH=/bin/ls:/etc/passwd:/usr/bin`
136. Which command line makes pathnames **/usr/local/bin** and **/usr/bin** lead to the same directory?
- `ln -s . /usr/local`
 - `mkdir /usr/local`
 - `ln . /usr/local`
 - `touch /usr/local`
 - `rmdir /usr/local`

154. What is the link count of file **a** after these successful commands?
ln a d ; cp a f ; ln d c ; ln f g ; ln c e
a. 5 b. 2 c. 4 d. 1 e. 3
155. If **/bin/xxx** is a program that outputs **one** and **/usr/bin/xxx** is a program that outputs **two**, what is the output on your screen after this:
PATH=/bin/xxx:/usr/bin/xxx:/etc/passwd ; xxx
a. **two** followed by **one**
b. **bash: xxx: command not found**
c. **one** followed by **two**
d. **two**
e. **one**
156. If your **PATH** variable contains **/bin:/usr/bin**, what is the output of this:
echo '\$PATH'
a. **/bin:/usr/bin**
b. **'\$PATH'**
c. **echo: \$PATH: No such file or directory**
d. **\$PATH**
e. **'/bin:/usr/bin'**
157. If you want a user-defined alias in all your **bash** shells, what do you do?
a. put the alias into the **/etc/group** file for next log in
b. create the alias and then type **save** to save it to all shells
c. put the alias into the **/etc/passwd** file for next log in
d. define the alias in my file **\$HOME/.bashrc**
e. put the alias into the **/bin/bash** file for next log in
158. In an empty directory, what is in file **foo** after this:
echo hi >foo ; ls nosuchfile | cat >foo
a. **hi**
b. nothing (empty file)
c. **nosuchfile**
d. **foo**
e. **ls: cannot access nosuchfile**
159. If the file **pig** contained the word **bar**, what is the output on your screen after this: **PATH=/etc/passwd:/bin/ls:/bin/who ; /bin/cat pig**
a. **bar**
b. **bash: /bin/cat: command not found**
c. **pig**
d. no output on screen
e. **/bin/cat: pig: No such file or directory**

160. If **/bin/foo** is a program that outputs **hi** and **/usr/bin/foo** is a program that outputs **mom** what is the output on your screen after this:
PATH=/etc:/usr/bin:/bin ; foo
a. **bash: foo: command not found**
b. **hi**
c. **hi** followed by **mom**
d. **mom** followed by **hi**
e. **mom**
161. If directory **/a** contains these seven two-character names: **aa, ab, ac, ad, a?, a*, a.**, then which command removes *only* the single two-character name **a?** from the directory?
a. **rm /a?** b. **rm /a/a*** c. **rm /a/a[*]**
d. **rm /a/a\?** e. **rm /a/a?**
162. What is the link count of file **f** after these successful commands?
cp f x ; ln f a ; ln x y ; ln a z ; ln a b
a. 6 b. 3 c. 5 d. 4 e. 2
163. How many arguments are passed to the command by the shell:
<bat bat -b "-a -r" >bat bat bat
a. 5 b. 3 c. 6 d. 7 e. 4
164. Your current directory is **dir1**. The parent directory contains another directory, **dir2**. Which command copies file **foo** from the current directory into the **dir2** directory?
a. **cp foo ../dir2** b. **cp foo dir2**
c. **cp foo dir2 ..** d. **cp .. dir2 foo**
e. **cp foo dir2/..**
165. If **mt** is an empty sub-directory, what is true after this:
touch foo ; mkdir bar ; mv foo bar/mt
a. the directory **mt** now contains a directory named **bar**
b. the command fails because **bar/mt** is not a directory
c. the directory **mt** now contains a file named **foo**
d. the directory **bar** now contains a file named **foo**
e. the directory **mt** is still empty
166. In an empty directory, what is the output on your screen after this:
echo one >bar ; echo .????*
a. **.????***
b. **.bar**
c. **.. .bar**
d. an error message from **echo** saying **.????*** does not exist
e. **one**

167. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
 111 -rw-r--r-- 1 me me 100 Jan 1 1:00 a
 222 -rw-r--r-- 1 me me 100 Jan 1 1:00 b
 333 -rw-r--r-- 1 me me 100 Jan 1 1:00 c
 444 -rw-r--r-- 2 me me 100 Jan 1 1:00 d
 a. 0 b. 1 c. 3 d. 4 e. 2
168. If `/bin/foo` is a program that outputs `dad` and `/usr/bin/foo` is a program that outputs `mom` what is the output on your screen after this:
`PATH=/usr:/etc:/bin:/usr/bin ; foo`
 a. `mom` followed by `dad`
 b. `dad` followed by `mom`
 c. `mom`
 d. `bash: foo: command not found`
 e. `dad`
169. What is the link count of directory `d` after these successful commands?
`mkdir d ; cd d ; touch a ; mkdir b c`
 a. 3 b. 2 c. 4 d. 6 e. 5
170. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `c` after this: `ln a d ; ln d c ; cp c b ; sort a b d >c`
 a. 5 b. 2 c. 6 d. 4 e. 0
171. Which command appends directory `/bin` to your search path?
 a. `PATH=$PATH:/bin` b. `$PATH=PATH:/bin`
 c. `PATH=PATH+/bin` d. `PATH=PATH:/bin`
 e. `$PATH=$PATH:/bin`
172. If `/bin/xxx` is a program that outputs `one` and `/usr/bin/xxx` is a program that outputs `two`, what is the output on your screen after this:
`PATH=/etc:/usr/bin:/usr:/bin ; /bin/xxx`
 a. `two`
 b. `one`
 c. `one` followed by `two`
 d. `two` followed by `one`
 e. `bash: /bin/xxx: command not found`
173. What is in file `c` after this:
`echo B >b ; ln b a ; echo A >a ; ln a c ; rm a b`
 a. `B` b. `A` followed by `B`
 c. `A` d. no such file (nonexistent)
 e. nothing (empty file)
174. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `c` after this: `ln a e ; ln b d ; ln d c ; cat e b >c`
 a. 3 b. 2 c. 0 d. 5 e. 4

175. Rewrite as a simplified absolute path:
`/usr/./bin/./lib/./../etc/./usr/./lib/./bin/./bar`
 a. `/usr/bin/bar` b. `/bar` c. `/usr/lib/bar`
 d. `/etc/bar` e. `/usr/bar`
176. What is the link count of directory `z` after these successful commands?
`mkdir z ; cd z ; touch a ; ln a b ; ln a c`
 a. 1 b. 5 c. 4 d. 3 e. 2
177. What is the link count of directory `x` after these successful commands?
`mkdir x ; mkdir x/y ; mkdir x/z ; mkdir x/y/z`
 a. 2 b. 3 c. 5 d. 4 e. 1
178. If `/bin/foo` is a program that outputs `mom` and `/usr/bin/foo` is a program that outputs `dad` what is the output on your screen after this:
`PATH=/bin/foo:/usr/bin/foo:/usr ; foo`
 a. `mom` followed by `dad`
 b. `mom`
 c. `dad` followed by `mom`
 d. `bash: foo: command not found`
 e. `dad`
179. What is the output on your screen after this:
`PATH=/bin/cat:/bin/sh:/bin/ls ; ls nosuchfile`
 a. `bash: /bin/sh: No such file or directory`
 b. `bash: ls: command not found`
 c. `ls: nosuchfile: No such file or directory`
 d. `bash: /bin/ls: command not found`
 e. `ls: /bin/ls: command not found`
180. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `c` after this: `ln a d ; ln d c ; cat a b >c`
 a. 2 b. 4 c. 3 d. 0 e. 5
181. Which command line below shows only lines 6-10 of file `foo`?
 a. `tail -10 foo | head -6` b. `head -10 foo | tail -5`
 c. `head -10 foo | tail -6` d. `tail -15 foo | head -5`
 e. `head -6 foo | tail -10`
182. In an empty directory, what is the output on your screen after this:
`touch 1 2 3 ; cow="*" ; echo '$cow'`
 a. `*` b. `1 2 3` c. `'$cow'`
 d. `'1 2 3'` e. `$cow`
183. How many arguments are passed to the command by the shell:
`<bar bar -b "-a" '-r' >bar bar bar`
 a. 3 b. 6 c. 4 d. 7 e. 5

184. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this: `ln a e ; ln b d ; ln d c ; cp d e ; sort a b e d >c`
 a. 6 b. 4 c. 12 d. 10 e. 7
185. In an empty directory, what is in file **count** after this:
`ls ??? | wc -w >count`
 a. nothing (empty file) b. 1
 c. 1 1 1 d. 1 1 2
 e. 0
186. If `/bin/bat` is a program that outputs **foo** and `/usr/bin/bat` is a program that outputs **hi** what is the output on your screen after this:
`PATH=/usr:/usr/bin:/bin ; bat`
 a. `bash: bat: command not found`
 b. **foo**
 c. **foo** followed by **hi**
 d. **hi**
 e. **hi** followed by **foo**
187. What is the link count of directory **dir** after these successful commands?
`mkdir dir ; cd dir ; touch foo ; mkdir a b c`
 a. 1 b. 4 c. 5 d. 2 e. 3
188. How many arguments are passed to the command by the shell:
`<bar bar -b"-a '-r' >bar" bar >out`
 a. 5 b. 3 c. 6 d. 2 e. 4
189. If directory **dir** contains only these five two-character names: **a?**, **11**, **?1**, **1***, **.1**, then which command removes *only* the single two-character name **?1** from the directory?
 a. `rm dir/\??` b. `rm dir/??` c. `rm dir/1*`
 d. `rm dir/?1` e. `rm dir/*1`
190. Which command shows names under directory **oldnotes** containing **RTFM** anywhere in the name?
 a. `ls oldnotes/RTFM*` b. `ls oldnotes RTFM *`
 c. `ls oldnotes*RTFM*` d. `ls oldnotes *RTFM*`
 e. `ls oldnotes/*RTFM*`
191. What is the link count of directory **dir** after these successful commands?
`mkdir dir ; cd dir ; touch one ; mkdir two`
 a. 5 b. 1 c. 2 d. 4 e. 3

192. If `/bin/foo` is a program that outputs **one** and `/usr/bin/foo` is a program that outputs **two**, what is the output on your screen after this:
`PATH=/dev:/usr/bin:/usr:/bin:/etc ; /bin/foo`
 a. **one** followed by **two**
 b. **two** followed by **one**
 c. **two**
 d. `bash: /bin/foo: command not found`
 e. **one**
193. If **mt** is an empty sub-directory, what is true after this:
`touch bar ; mkdir foo ; mv mt/../../bar mt/foo`
 a. the directory **mt** now contains a file named **bar**
 b. the command fails because `mt/foo` is not a directory
 c. the directory **mt** now contains a file named **foo**
 d. the directory **foo** now contains a file named **bar**
 e. the directory **mt** is still empty
194. What is the link count of directory **dir** after these successful commands?
`mkdir dir ; mkdir dir/foo ; touch dir/bar`
 a. 5 b. 3 c. 1 d. 2 e. 4
195. What is the link count of file **f** after these successful commands?
`rm f ; touch f ; ln f b ; cp f g
cp b a ; ln a d ; ln b c ; cp c g`
 a. 4 b. 2 c. 5 d. 1 e. 3
196. If directory **/a** contains these seven two-character names: **aa**, **ab**, **ac**, **ad**, **a?**, **a***, **a.**, then which command removes *only* the single two-character name **a*** from the directory?
 a. `rm /a*` b. `rm /a/a*` c. `rm /a/*`
 d. `rm "/a/a"` e. `rm /a/a?`
197. How many arguments are passed to the command by the shell:
`echo 'It's a bird! No! It's a plane!'`
 a. 2 b. 3 c. 5 d. 1 e. 4
198. Which file is a DOS/Windows file?
 a. ASCII text, with CRLF line terminators
 b. ASCII text
 c. ASCII text, with LF line terminators
 d. ASCII text, with no line terminators
 e. ASCII text, with CR line terminators
199. If you are in **/bin** and `ls -l` shows a symbolic link **bar** `-> ../dir/foo` then dereference the absolute path of **bar** with no symbolic links:
 a. `/bin/bar/dir/foo` b. `/bin/dir/foo/bar`
 c. `/bin/dir/foo` d. `/bar/../../dir/foo`
 e. `/dir/foo`

200. If you are in `/bin` and `ls -l` shows a symbolic link `foo -> dir/bar` then dereference the absolute path of `foo` with no symbolic links:
- `/foo/dir/bar`
 - `/bin/dir/bar`
 - `/bin/foo/dir/bar`
 - `/dir/bar`
 - `/bin/dir/bar/foo`
201. Which command line shows the current date?
- `echo date | bash`
 - `bash date`
 - `date | bash`
 - `bash <date`
 - `bash >date ; cat date`
202. How many files are touched? `touch "1" "2 3" " " ' ' 4 5`
- 3
 - 4
 - 6
 - 5
 - 7
203. If directory `/a` contains these seven two-character names: `aa, ab, ac, ad, a*, a?, ??`, then which command removes *only* the single two-character name `a?` from the directory?
- `rm /a\?`
 - `rm "/a?"`
 - `rm '/a/a?'`
 - `rm /a/?\?`
 - `rm /a/a?`
204. If `/bin/pig` is a program that outputs `hi` and `/usr/bin/pig` is a program that outputs `foo` what is the output on your screen after this:
- ```
PATH=/etc:/usr/bin:/bin ; pig
```
- `hi` followed by `foo`
  - `foo`
  - `hi`
  - `foo` followed by `hi`
  - `bash: pig: command not found`
205. How many arguments are passed to the command by the shell:
- ```
echo " 1 '2 3' 4 "5 6 ' 7 "8 ' >out
```
- 6
 - 4
 - 3
 - 2
 - 5
206. Which command counts lines containing the string `refused` in only the month `October 2016` in the `denyhosts` log file?
- `fgrep -c 2016-10 denyhosts | fgrep refused`
 - `fgrep 'refused' denyhosts | fgrep -c '2016-10'`
 - `fgrep '2016-10 refused' denyhosts`
 - `fgrep refused denyhosts | fgrep -c October 2016`
 - `fgrep refused denyhosts ; fgrep -c 2016-10`
207. What is true about this output from `ls -il foo bar`?
- ```
15 -rwxrwxrwx 2 bin bin 3 Jul 31 12:33 foo
15 -rwxrwxrwx 3 bin bin 3 Jul 31 12:33 bar
```
- `foo` and `bar` are two of three names for the same file
  - this output is not possible
  - `foo` and `bar` are names for different files
  - `foo` and `bar` are names for the same file
  - `foo` and `bar` each have three names (six names total)

208. What is the link count of directory `a` after these successful commands?
- ```
mkdir a ; mkdir a/b ; mkdir a/c ; mkdir a/b/c
```
- 5
 - 4
 - 3
 - 2
 - 1
209. What displays on your screen given this command:
- ```
date >date ; pwd >pwd ; head date | tail pwd
```
- `head` displays the `date` and `tail` displays the `pwd`
  - only the `date` displays because `tail` ignores the pipe
  - nothing displays because `tail` ignores the pipe
  - `tail` reads the pipe and the `pwd` and displays both together
  - only the `pwd` displays because `tail` ignores the pipe
210. What is the output of this in an empty directory:
- ```
touch 1 13 .13 2 213 3 30 39 .31 ; echo [13]?
```
- an error message from `echo` saying `[13]?` does not exist
 - `1 13 3 30 39`
 - `13`
 - `[13]?`
 - `13 30 39`
211. If `foo` were a readable empty file, what is the output on your screen after this:
- ```
PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/cat foo
```
- no output on screen
  - `bash: ls: command not found`
  - `bash: cat: command not found`
  - `/bin/cat: foo: No such file or directory`
  - `bash: /bin/cat: command not found`
212. How many arguments are passed to the command by the shell:
- ```
<f z " a 'b c' d " 1 2 ' g " h " ' >z
```
- 5
 - 4
 - 3
 - 6
 - 2
213. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `d` (not in `c`) after this:
- ```
ln a d ; ln d c ; ln c e ; cat a a b b c c d d e e >c
```
- 2
  - 6
  - 10
  - 18
  - 21
214. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
- ```
111 -rw-r--r-- 2 me me 1 Jan 1 1:00 a
111 -rw-r--r-- 2 me me 1 Jan 1 1:00 b
222 -rw-r--r-- 3 me me 1 Jan 1 1:00 c
222 -rw-r--r-- 3 me me 1 Jan 1 1:00 d
```
- 0
 - 4
 - 2
 - 1
 - 3
215. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `c` after this: `ln a d ; ln d e ; ln b f >c`
- 4
 - 3
 - 2
 - 5
 - 0

216. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this: `sort a b >c ; cat a >>b ; cat c b >c a`
 a. 0 b. 7 c. 12 d. 8 e. 5
217. How many arguments are passed to the command by the shell:
`<foo foo " a 'b c' d " e ' f " g " ' >foo h`
 a. 6 b. 2 c. 4 d. 3 e. 5
218. Which command line shows just the count of words in the file?
 a. `wc file | awk '{print #2}'`
 b. `wc file | awk '[print $2]'`
 c. `wc file | awk '[print #2]'`
 d. `wc file | awk '{print 2}'`
 e. `wc file | awk '[print $2]'`
219. In an empty directory, what is in file **out** after this:
`ls nosuchfile | wc -w >out`
 a. 1 b. **out**
 c. nothing (empty file) d. **nosuchfile**
 e. 0
220. If directory **/a** contains these seven two-character names: **aa, ab, ac, ad, a?, a*, a.**, then which command removes *only* the single two-character name **a*** from the directory?
 a. `rm /a*` b. `rm /a/a?` c. `rm /a/a*`
 d. `rm /a/*` e. `rm /a/a*`
221. How do you execute the program **foo** in the current directory?
 a. `foo/` b. `/foo` c. `$HOME/foo`
 d. `foo/.` e. `./foo`
222. In an empty directory, what is the output on your screen after this:
`touch 1 2 3 ; cow="*" ; echo $cow`
 a. "1 2 3" b. 1 2 3 c. *
 d. \$cow e. "*" e. *
223. In an empty directory, what is the output on your screen after this:
`echo hi >foo ; cp foo bar | wc -w`
 a. 3 b. 1 c. 2
 d. 0 e. no output
224. If **mt** is an empty sub-directory, what is true after this:
`touch mt/bar ; mkdir bar ; mv mt/bar mt/./bar/me`
 a. there is a second copy of the file **bar** in the file named **me**
 b. the directory **mt** is now empty
 c. the command fails because the name `mt/./bar/me` does not exist
 d. the `mkdir` fails because **bar** already exists
 e. the directory **mt** now contains only a file named **me**

225. What is true about this output from `ls -il foo bar`
`15 -r-x----- 2 me me 3 Jan 1 1:00 foo`
`99 -r-x----- 2 me me 3 Jan 1 1:00 bar`
 a. **foo** and **bar** are two of three names for the same file
 b. **foo** and **bar** each have three names (six names total)
 c. **foo** and **bar** are names for the same file
 d. this output is not possible
 e. **foo** and **bar** are names for different files
226. Which command counts lines with two adjacent asterisk characters (******) inside the file?
 a. `find file -name **` b. `find -c ** file`
 c. `fgrep -c ** file` d. `fgrep -c '**' file`
 e. `find file -name '**'`
227. What is the link count of directory **z** after these successful commands?
`mkdir z ; mkdir z/a z/a/b z/a/c z/a/d`
 a. 1 b. 5 c. 4 d. 3 e. 2
228. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this: `ln a d ; ln b e ; cp d e >c`
 a. 2 b. 0 c. 3 d. 4 e. 5
229. What is in file **out** after this:
`echo me >a ; ln a b ; echo hi >b ; ln a out ; rm a b`
 a. **me** followed by **hi** b. nothing (empty file)
 c. no such file (nonexistent) d. **hi**
 e. **me**
230. File **a** contains 3 lines. File **b** contains 4 lines. How many lines are output on your screen by this: `sort a | echo b`
 a. 1 b. 3 followed by 4 c. 3 followed by 1
 d. 3 e. 4
231. In an empty directory, what is the output on your screen after this:
`echo one >.bar ; echo .*`
 a. **.bar**
 b. **one**
 c. **. .. .bar**
 d. an error message from `echo` saying **.*** does not exist
 e. **.***
232. What is the output on your screen after these command lines:
`echo one >x ; ln x y ; echo two >y`
`echo ten >x ; cat y`
 a. **one** b. **ten**
 c. **one** followed by **two** and **ten** d. no output on screen
 e. **two**

233. What displays on your screen given this command:

```
ls >ls ; wc ls >wc ; sort ls | cat wc
```

- a. only the **wc** displays because **cat** ignores the pipe
- b. only the **ls** displays because **cat** ignores the pipe
- c. **cat** reads the pipe and the **wc** and displays both together
- d. **sort** displays the **ls** and **cat** displays the **wc**
- e. nothing displays because **cat** ignores the pipe

234. Did you read all the words of the test instructions on page one?

- a. **Tak** (*Yes - Polish*)
- b. **Igen** (*Yes - Hungarian*)
- c. **Sim** (*Yes - Portuguese*)
- d. **Jes** (*Yes - Esperanto*)
- e. **Taip** (*Yes - Lithuanian*)

This page intentionally left blank.