

PRINT Name: \_\_\_\_\_ LAB Section:

**One-Answer Multiple Choice 233 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 **B**  
 192. Answer **192** is **A**  
 193. Answer **193** is **E**  
 194. Answer **194** is **A**  
 195. Answer **195** is **B**  
 196. Answer **196** is **B**

Your Test Version is:

**B A E A B B**

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

1. **Did you read all the words of the test instructions on page one?**
  - a. **Sim** (Yes - Portuguese)
  - b. **Tak** (Yes - Polish)
  - c. **Jes** (Yes - Esperanto)
  - d. **Igen** (Yes - Hungarian)
  - e. **Taip** (Yes - Lithuanian)
2. **My three-digit Lab Section number is:**
  - a. My lecture section number, e.g. **010** or **020**.
  - b. My lab room number, e.g. **B384**, **T321**, **J218**, **N201**
  - c. My lecture room number, e.g. **T130**, **T117**
  - d. The timetable section number of my weekly 2-hour lab period.
  - e. The Test Version number printed in the top left corner.
3. The correct syntax to assign to a shell variable is:
  - a. **V=foo bar**
  - b. **V="foo bar"**
  - c. **V = foo bar**
  - d. **"V=foo bar"**
  - e. **V = "foo bar"**
4. 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. 1
  - b. 4
  - c. 5
  - d. 3
  - e. 2
5. In an empty directory, how many words are in file **c** after this:
 

```
touch a ; mv b a >b ; ls >c
```

  - a. 0
  - b. 2
  - c. 1
  - d. 3
  - e. 4

6. In an empty directory, what is the output on your screen after this:
 

```
echo one >.bar ; echo .????*
```

  - a. **.. .bar**
  - b. **.????\***
  - c. **.bar**
  - d. an error message from **echo** saying **.????\*** does not exist
  - e. **one**
7. 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. put the alias into the **/etc/passwd** file for next log in
  - c. create the alias and then type **save** to save it to all shells
  - d. put the alias into the **/bin/bash** file for next log in
  - e. define the alias in my file **\$HOME/.bashrc**
8. 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]'**
9. 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. **one** followed by **two** and **ten**
  - c. no output on screen
  - d. **ten**
  - e. **two**
10. 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 directory **mt** is still empty
  - c. the directory **bar** now contains a file named **foo**
  - d. the directory **mt** now contains a file named **foo**
  - e. the command fails because **bar/mt** is not a directory
11. 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** are names for different files
  - c. **foo** and **bar** are names for the same file
  - d. **foo** and **bar** each have three names (six names total)
  - e. this output is not possible

12. In an empty directory, what is the output on your screen after this:  
`echo one >.bar ; echo .??*`
- one
  - .??\*
  - an error message from `echo` saying `.??*` does not exist
  - `.. .bar`
  - `.bar`
13. How do you execute the program `foo` in the current directory?
- `/foo`
  - `./foo`
  - `foo/.`
  - `foo/`
  - `$HOME/foo`
14. What is the output on your screen after these command lines:  
`echo one >x ; ln x y ; echo ten >y`  
`echo two >x ; cat y`
- one followed by `ten` and `two`
  - `ten`
  - `two`
  - one
  - no output on screen
15. What is in file `out` after this:  
`echo me >a ; ln a b ; echo hi >b ; ln a out ; rm a b`
- `me` followed by `hi`
  - no such file (nonexistent)
  - `hi`
  - nothing (empty file)
  - `me`
16. Which command recursively finds all things named `foo`?
- `find -name foo`
  - `fgrep -name foo`
  - `ls -name foo`
  - `cat -name foo`
  - `echo -name foo`
17. If directory `/a` contains these seven two-character names: `aa`, `ab`, `ac`, `ad`, `a?`, `a*`, `a.`, then which command removes *only* the single two-character name `a?` from the directory?
- `rm /a/a\?`
  - `rm /a/a*`
  - `rm /a/a[*]`
  - `rm /a/a?`
  - `rm /a/.`
18. In an empty directory, what is the output on your screen after this:  
`touch 1 2 3 ; cow="" ; echo '$cow'`
- `'$cow'`
  - `*`
  - `'1 2 3'`
  - `1 2 3`
  - `$cow`
19. 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 32 is the count of links (names) this directory has.
  - The number 128 is the size of this directory.
  - The number 32 is the inode number of this directory.
  - The number 128 is the count of links (names) this directory has.
  - The number 128 is the inode number of this directory.

20. Which command removes *only* this four-character name containing a special character: `*xyz`
- `rm "*"xyz"`
  - `rm *xyz`
  - `rm '*xyz'`
  - `rm "*xyz"`
  - `rm '*xyz''`
21. Which environment variable contains your HOME directory?
- `$home`
  - `$/HOME`
  - `/home/abcd0001`
  - `$HOME`
  - `/home`
22. If your `PATH` contained only the file names `/bin/sh`, `/bin/cat`, and `/bin/ls`, then what is the output on your screen of this command:  
`cat /etc/passwd`
- `cat: bash: no such file or directory`
  - `cat: /etc/passwd: command not found`
  - `bash: cat: command not found`
  - `bash: /bin/cat: no such file or directory`
  - `bash: /bin/sh: command not found`
23. If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
- ```
111 -rw-r--r-- 2 me me 100 Jan 1 1:00 a
222 -rw-r--r-- 2 me me 100 Jan 1 1:00 b
333 -rw-r--r-- 2 me me 100 Jan 1 1:00 c
444 -rw-r--r-- 2 me me 100 Jan 1 1:00 d
```
- 3
  - 2
  - 0
  - 4
  - 1
24. If your `PATH` variable contains `/bin:/usr/bin`, what is the output of this:  
`echo '$PATH'`
- `/bin:/usr/bin`
  - `echo: $PATH: No such file or directory`
  - `'$PATH'`
  - `'/bin:/usr/bin'`
  - `$PATH`
25. How many arguments are passed to the command by the shell:  
`<bar bar -b "-a" '-r' >bar bar bar`
- 7
  - 3
  - 4
  - 5
  - 6
26. 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/b1 a/b2`
  - no output
  - `b1 b2`
  - `b a/b1 a/b2`
  - `b`

27. 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**
- no output on screen
  - bar**
  - /bin/cat: pig: No such file or directory**
  - pig**
  - bash: /bin/cat: command not found**
28. Which option to **ls** displays the directory itself and not its contents?
- R**
  - d**
  - i**
  - l**
  - a**
29. Which of the following **PATH** statements makes the most sense?
- PATH=/bin/ls:/etc/passwd:/usr/bin**
  - PATH=/bin:/etc/passwd:/usr/bin**
  - PATH=/bin/bash:/usr/bin:/bin**
  - PATH=/bin:/usr/bin:/etc/passwd**
  - PATH=/bin:/usr/bin**
30. 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
  - 3
  - 2
  - 1
  - 4
31. 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
```
- two**
  - bash: /bin/xxx: command not found**
  - one** followed by **two**
  - two** followed by **one**
  - one**
32. 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
```
- hi**
  - bash: prg: command not found**
  - foo** followed by **hi**
  - hi** followed by **foo**
  - foo**

33. If **mt** is an empty sub-directory, what is true after this:
- ```
touch mt/bar ; mkdir bar ; mv mt/bar mt/./bar/me
```
- the **mkdir** fails because **bar** already exists
  - the directory **mt** now contains only a file named **me**
  - the directory **mt** is now empty
  - the command fails because the name **mt/./bar/me** does not exist
  - there is a second copy of the file **bar** in the file named **me**
34. In an empty directory, what is the output on your screen after this:
- ```
touch A a ; echo * >"*" ; ls
```
- \* >\***
  - \* A a**
  - A a >\***
  - A a >A a**
  - No output
35. How many files are touched? **touch 1 "2 3 ' 4 '" 5**
- 3
  - 4
  - 2
  - 5
  - 1
36. How many arguments are passed to the command by the shell:
- ```
<wc wc " 1 '2 3' 4 " 5 6 ' 7 " 8 " ' >wc 9
```
- 4
  - 5
  - 2
  - 3
  - 6
37. What is the output on your screen after this:
- ```
echo one >x ; ln x y ; echo two >>y ; sort x
```
- two** followed by **one**
  - one** followed by **two**
  - no output
  - two**
  - one**
38. Rewrite as a simplified absolute path:
- ```
/home/me/./you/../../../../etc/./home/me/./you/./me/./foo
```
- /etc/foo**
  - /home/me/foo**
  - /home/foo**
  - /foo**
  - /home/you/foo**
39. 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**
- foo**
  - bat**
  - bash: cat: command not found**
  - no output on screen
  - cat: bat: No such file or directory**
40. Which command line below shows only lines 6-10 of file **foo**?
- tail -15 foo | head -5**
  - head -10 foo | tail -5**
  - head -6 foo | tail -10**
  - head -10 foo | tail -6**
  - tail -10 foo | head -6**
41. What is the link count of directory **foo** after these successful commands?
- ```
mkdir foo ; cd foo ; touch a b c
```
- 2
  - 3
  - 4
  - 1
  - 5

42. 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`
- `dad`
  - `mom`
  - `bash: foo: command not found`
  - `mom` followed by `dad`
  - `dad` followed by `mom`
43. 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?
- `rm dir/*1`
  - `rm dir/??`
  - `rm dir/?1`
  - `rm dir/1*`
  - `rm dir/\??`
44. 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`
- 3
  - 9
  - 6
  - 2
  - 5
45. In an empty directory, what is the output on your screen after this:  
`touch 1 2 3 ; cow="*" ; echo "$cow"`
- `*`
  - `"$cow"`
  - `1 2 3`
  - `$cow`
  - `"1 2 3"`
46. 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
```
- `foo` and `bar` each have two names (four names total)
  - `foo` and `bar` each have three names (six names total)
  - `foo` and `bar` are names for the same file
  - `foo` and `bar` are two of three names for this file
  - this output is not possible
47. What is the link count of directory `dir` after these successful commands?  
`mkdir dir ; mkdir dir/foo ; touch dir/bar`
- 4
  - 2
  - 5
  - 3
  - 1
48. 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/a\*`
  - `rm /a/a*`
  - `rm /a/a?`
  - `rm /a/*`
  - `rm /a*`

49. 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`
- `bash: bat: command not found`
  - `foo` followed by `hi`
  - `hi` followed by `foo`
  - `hi`
  - `foo`
50. 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`
- 3
  - 2
  - 1
  - 4
  - 0
51. What displays on your screen given this command:  
`date >date ; pwd >pwd ; head date | tail pwd`
- `tail` reads the pipe and the `pwd` and displays both together
  - only the `pwd` displays because `tail` ignores the pipe
  - `head` displays the `date` and `tail` displays the `pwd`
  - nothing displays because `tail` ignores the pipe
  - only the `date` displays because `tail` ignores the pipe
52. What is the link count of directory `dir` after these successful commands?  
`mkdir dir ; touch foo ; cd dir ; ln ../foo bar`
- 2
  - 5
  - 3
  - 1
  - 4
53. What is the output on your screen after this:  
`echo hi >a ; cp a b | wc -w`
- 3
  - 0
  - 1
  - 2
  - no output
54. You enter this `cp a/b c/` and get `cp: a: No such file or directory` because:
- you forgot to specify the destination file name after `c/`
  - directory `a` does not exist
  - the command `cp` is not in your search `PATH`
  - pathname `a` exists but is a file, not a directory
  - directory `c` does not exist
55. A "dangling symlink" is a symlink to:
- a parent directory
  - a non-existent target
  - the current directory
  - a directory
  - a special device file
56. 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
  - 5
  - 1
  - 3
  - 2

57. What is in the local variable `$$` ?
- the first argument of the previous command line
  - the cpu cost of the current session, in dollars
  - the command name of the previous command line
  - the process ID of the current shell
  - `$$` is not a valid variable name
58. What is the link count of directory `dir` after these successful commands?
- ```
mkdir dir ; cd dir ; touch a b c ; mkdir d e
```
- 3
  - 2
  - 7
  - 4
  - 5
59. 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
```
- 1
  - 4
  - 3
  - 5
  - 2
60. How many arguments are passed to the command by the shell:
- ```
<f z " a 'b c' d " 1 2 ' g " h " ' >z
```
- 5
  - 3
  - 4
  - 6
  - 2
61. What is the link count of directory `d` after these successful commands?
- ```
mkdir d ; mkdir d/a d/b ; touch d/c d/e
```
- 3
  - 1
  - 2
  - 4
  - 5
62. How many files are touched? `touch "1 " 2 3" " ' ' 4 5`
- 7
  - 5
  - 4
  - 3
  - 6
63. 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
```
- `one` followed by `two`
  - `two` followed by `one`
  - `two`
  - `bash: xxx: command not found`
  - `one`
64. How many arguments are passed to the command by the shell:
- ```
<cow cow "-x" -y '-z' >cow cow
```
- 4
  - 6
  - 5
  - 2
  - 3
65. Which one of these names is usually a shell environment variable?
- `fooBar`
  - `foobar`
  - `foobar`
  - `FooBar`
  - `FOOBAR`
66. What is usually in the environment variable `$$SHELL`?
- the absolute path of your login shell
  - the relative path of the system `/shell` directory
  - the relative path of the `/home/shell` directory
  - the absolute path of the system `/shell` directory
  - the relative path of your login shell

67. What is the link count of file `f` after these successful commands?
- ```
rm f ; touch f ; ln f a ; ln a b
cp f c ; ln c x ; rm b ; mv a b
```
- 4
  - 2
  - 0
  - 3
  - 1
68. What displays on your screen given this command:
- ```
ls >ls ; wc ls >wc ; sort ls | cat wc
```
- only the `ls` displays because `cat` ignores the pipe
  - `sort` displays the `ls` and `cat` displays the `wc`
  - `cat` reads the pipe and the `wc` and displays both together
  - only the `wc` displays because `cat` ignores the pipe
  - nothing displays because `cat` ignores the pipe
69. 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
```
- 2
  - 2 followed by 1
  - 1 followed by 2
  - no output
  - 1
70. In an empty directory, what is in file `out` after this:
- ```
ls nosuchfile | wc -w >out
```
- nothing (empty file)
  - `out`
  - 0
  - 1
  - `nosuchfile`
71. In an empty directory, what is the output on your screen after this:
- ```
echo hi >foo ; cp foo bar | wc -w
```
- 0
  - 3
  - no output
  - 2
  - 1
72. 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:
- `/bin/bar/foo`
  - `/bar`
  - `/bin/bar`
  - `/foo/bar`
  - `/bin/foo/bar`
73. What is the link count of file `f` after these successful commands?
- ```
rm f ; touch f ; ln f b ; cp f c
cp b x ; ln x y ; ln b z ; ln z a
```
- 5
  - 1
  - 2
  - 4
  - 3
74. What is in file `c` after this:
- ```
echo B >b ; ln b a ; echo A >a ; ln a c ; rm a b
```
- nothing (empty file)
  - `B`
  - `A` followed by `B`
  - `A`
  - no such file (nonexistent)
75. 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
```
- 1
  - 3
  - 5
  - 2
  - 4

76. 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`
- `foo` followed by `hi`
  - `hi`
  - `foo`
  - `bash: pig: command not found`
  - `hi` followed by `foo`
77. What is the output of this in an empty directory:  
`date >.date ; users >.users ; echo .?*`
- `.date`
  - an error message from `echo` saying `.?*` does not exist
  - `.. .date .users`
  - `.date .users`
  - `.?*`
78. In an empty directory, what is the output on your screen after this:  
`echo one >.bar ; ls .????*`
- `.????*`
  - `one`
  - an error message from `ls` saying `.????*` does not exist
  - `.. .bar`
  - `.bar`
79. In an empty directory, what is the output on your screen after this:  
`echo hi >a ; ls | wc -w`
- 2
  - 0
  - 1
  - no output
  - a
80. What command will recursively show disk usage in directories?
- `tree`
  - `ls`
  - `df`
  - `du`
  - `find`
81. 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:
- `/etc/foo/bar`
  - `/bar/foo`
  - `/etc/bar/foo`
  - `/foo`
  - `/etc/foo`
82. 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?
- `cp foo dir2 ..`
  - `cp foo dir2`
  - `cp .. dir2 foo`
  - `cp foo dir2/..`
  - `cp foo ../dir2`

83. In an empty directory, what is in file `count` after this:  
`ls ??? | wc -w >count`
- 1 1 2
  - 1 1 1
  - 1
  - 0
  - nothing (empty file)
84. Which command line would show the index (inode) number of a file?
- `cat -i file`
  - `find -i file`
  - `ls -l file`
  - `cat -l file`
  - `ls -i file`
85. 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`
- 3
  - 2
  - 4
  - 1
  - 5
86. What is the link count of directory `a` after these successful commands?  
`mkdir a ; mkdir a/b ; mkdir a/c ; mkdir a/b/c`
- 3
  - 1
  - 5
  - 2
  - 4
87. In an empty directory, what is the output on your screen after this:  
`touch a ; ls | wc -w`
- 1
  - 0
  - no output
  - 3
  - 2
88. What command will recursively find all pathnames named `foo` in `/bin`?
- `grep /bin -basename 'foo'`
  - `ls -R 'foo' /bin`
  - `find /bin -name 'foo'`
  - `grep 'foo' /bin`
  - `find foo -name '/bin'`
89. How many arguments are passed to the command by the shell:  
`echo "cow "y " bat 'man x' " pig'a "hop' a b`
- 6
  - 5
  - 7
  - 11
  - 4
90. What is the link count of directory `x` after these successful commands?  
`mkdir x ; mkdir x/y ; mkdir x/z ; mkdir x/y/z`
- 2
  - 3
  - 4
  - 1
  - 5
91. In an empty directory, how many words are in file `c` after this:  
`touch a b 1 b a ; ls >c`
- 0
  - 4
  - 2
  - 3
  - 1
92. 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-- 3 me me 100 Jan 1 1:00 c
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 d
```
- 3
  - 4
  - 1
  - 0
  - 2

93. Which command counts lines with two adjacent asterisk characters (\*\*) inside the file?
- a. `fgrep -c ** file`                      b. `find -c ** file`  
 c. `fgrep -c '**' file`                    d. `find file -name '**'`  
 e. `find file -name **`
94. Which command copies a directory:
- a. `mv -rf dir1 dir2`                      b. `mv -r dir1 dir2`  
 c. `cp -r dir1 dir2`                      d. `mv -f dir1 dir2`  
 e. `cp dir1 dir2`
95. 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
```
- a. 5                      b. 4                      c. 3                      d. 1                      e. 2
96. 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. 3                      b. 4                      c. 1                      d. 0                      e. 2
97. 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`
98. The option to `ls` that shows inode (index) numbers is:
- a. `-i`                      b. `-l`                      c. `-x`                      d. `-1`                      e. `-a`
99. 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`
- a. 3                      b. 5                      c. 0                      d. 2                      e. 4
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. 2                      b. 4                      c. 6                      d. 3                      e. 5
101. Rewrite as a simplified absolute path:
- ```
../../../../var/./a/../../../../var/b/../../../../etc/./bar/./foo
```
- a. `/var/foo`                      b. `/var/b/foo`                      c. `/etc/foo`  
 d. `/var/a/foo`                      e. `/etc/bar/foo`

102. 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. 3                      d. 0                      e. 2
103. How many arguments are passed to the command by the shell:
- ```
echo " one '2 three' 4 "five 6 ' 7 "8 ' >out
```
- a. 3                      b. 5                      c. 4                      d. 2                      e. 6
104. What is the output on your screen after this:
- ```
mkdir foo ; rmdir foo | wc -w
```
- a. 2                      b. 3                      c. 1  
 d. no output                      e. 0
105. 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. `foo` and `bar` are names for different files  
 b. `foo` and `bar` are names for the same file  
 c. `foo` and `bar` each have three names (six names total)  
 d. `foo` and `bar` are two of three names for the same file  
 e. this output is not possible
106. What is in file `c` after this:
- ```
echo A >a ; ln a b ; echo B >b ; ln a c ; rm a b
```
- a. `B`                      b. `A`  
 c. `A` followed by `B`                      d. no such file (nonexistent)  
 e. nothing (empty file)
107. Which command line allows programs in the current directory to execute without preceding the names with `./`? (P.S. Security Risk! Don't do this!)
- a. `PATH=./$HOME:/usr/bin`                      b. `PATH=/usr/bin/.$HOME`  
 c. `PATH=/bin:/usr/bin:.`                      d. `$PATH=/usr/bin:./bin`  
 e. `$PATH=.:$HOME:/usr/bin`
108. In an empty directory, how many words are in file `a` after this:
- ```
echo It's redirected >b isn't it\? ; ls >a
```
- a. 3                      b. 4                      c. 2                      d. 0                      e. 1
109. 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. 3                      b. 2                      c. 4                      d. 5                      e. 1

110. In an empty directory, what is in file **foo** after this:  
`echo hi >foo ; ls nosuchfile | cat >foo`  
 a. **hi**  
 b. **nosuchfile**  
 c. nothing (empty file)  
 d. **foo**  
 e. **ls: cannot access nosuchfile**
111. Which of these statements is true?  
 a. Only single quotes are strong enough to stop GLOB patterns from expanding.  
 b. If **/x** is an empty directory, `sort /x/*` produces an error message.  
 c. Only backslashes are strong enough to stop GLOB patterns from expanding.  
 d. If **/y** is an empty directory, `echo /y/*` produces an error message.  
 e. Only double quotes are strong enough to stop GLOB patterns from expanding.
112. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are output on your screen by this: `cat b | cat a`  
 a. **5**  
 b. **3**  
 c. **3 followed by 2**  
 d. **2 followed by 3**  
 e. **2**
113. What is the output on your screen after this:  
`PATH=/bin/cat:/bin/sh:/bin/ls ; ls nosuchfile`  
 a. **bash: /bin/ls: command not found**  
 b. **ls: /bin/ls: command not found**  
 c. **bash: ls: command not found**  
 d. **ls: nosuchfile: No such file or directory**  
 e. **bash: /bin/sh: No such file or directory**
114. To change to the parent directory, do this:  
 a. `pwd ..`  
 b. `pwd`  
 c. `cd ..`  
 d. `cd .`  
 e. `cd`
115. 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. no output  
 b. **2**  
 c. **a**  
 d. **0**  
 e. **1**
116. Create a symbolic link under **/usr** named **bar** that has target **xy**:  
 a. `ln -s 'xy' /bar/usr`  
 b. `ln -s /usr/bar 'xy'`  
 c. `ln -s 'xy' '/usr/bar'`  
 d. `ln -s /usr/bar '/usr/xy'`  
 e. `ln -s '/usr/xy' /usr/bar`
117. 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. **3**  
 b. **1**  
 c. **5**  
 d. **2**  
 e. **4**

118. How many arguments are passed to the command by the shell:  
`<bar bar -b"-a '-r' >bar" bar >out`  
 a. **6**  
 b. **5**  
 c. **2**  
 d. **3**  
 e. **4**
119. 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. **4**  
 b. **0**  
 c. **1**  
 d. **2**  
 e. **3**
120. How many arguments are passed to the command by the shell:  
`echo " 1 2 "three ' 4 ' five"6"`  
 a. **4**  
 b. **9**  
 c. **1**  
 d. **5**  
 e. **3**
121. What is the link count of directory **d** after these successful commands?  
`mkdir d ; touch f ; cd d ; ln ../f x`  
 a. **1**  
 b. **5**  
 c. **4**  
 d. **3**  
 e. **2**
122. 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. **bash: foo: command not found**  
 d. **two**  
 e. **one**
123. How many files are touched? `touch '1 "2 3 '4"' ' 5`  
 a. **1**  
 b. **4**  
 c. **3**  
 d. **5**  
 e. **2**
124. 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. **10**  
 b. **7**  
 c. **12**  
 d. **6**  
 e. **4**
125. 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. **0**  
 b. **3**  
 c. **4**  
 d. **1**  
 e. **2**



126. 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. `hi`  
 b. `hi` followed by `mom`  
 c. `mom`  
 d. `mom` followed by `hi`  
 e. `bash: foo: command not found`
127. How many arguments are passed to the command by the shell:  
`<cow cow "-x "-y '-z' >cow cow`  
 a. 6                    b. 4                    c. 7                    d. 3                    e. 5
128. 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. 4                    c. 3                    d. 5                    e. 7
129. 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. `/etc/bar/foo`                    b. `/etc/foo/bar`                    c. `/etc/foo`  
 d. `/bar/foo`                    e. `/foo`
130. 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. `mom` followed by `dad`  
 b. `mom`  
 c. `dad` followed by `mom`  
 d. `dad`  
 e. `bash: /bin/foo: command not found`
131. How many files are touched? `touch "1" 2 3 " " 4 5`  
 a. 5                    b. 4                    c. 6                    d. 3                    e. 7
132. 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. 4                    b. 1                    c. 3                    d. 2                    e. 0
133. Which command recursively finds all things with names beginning with `foo`?  
 a. `ls foo*`                    b. `find -name foo*`  
 c. `fgrep -name foo*`                    d. `find -name 'foo*'`  
 e. `ls -name foo*`
134. 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. 3                    b. 2                    c. 5                    d. 4                    e. 1

135. Dereference the following symlink `xyz` into its equivalent absolute path:  
`ln -s ../../a/./b/./bar /tmp/a/b/xyz`  
 a. `/tmp/b/bar`                    b. `/tmp/b/xyz`                    c. `/tmp/a/b/bar`  
 d. `/tmp/a/bar`                    e. `/tmp/bar`
136. In an empty directory, what is the output on your screen after this:  
`echo one >.bar ; echo .?*`  
 a. `.?*`  
 b. an error message from `echo` saying `.?* does not exist`  
 c. `.bar`  
 d. `one`  
 e. `.. .bar`
137. What is the output on your screen after these command lines:  
`echo one >x ; ln x y ; echo two >>y`  
`sort x >y ; cat y`  
 a. `two`                    b. `one` followed by `two`  
 c. no output                    d. `two` followed by `one`  
 e. `one`
138. What is the link count of directory `d` after these successful commands?  
`mkdir d ; cd d ; touch a ; mkdir b c`  
 a. 6                    b. 2                    c. 4                    d. 3                    e. 5
139. What is in file `foo` after this:  
`echo hi >a ; ln a b ; echo me >b ; ln a foo ; rm a b`  
 a. no such file (nonexistent)                    b. nothing (empty file)  
 c. `hi`                    d. `me`  
 e. `hi` followed by `me`
140. What is usually in the environment variable `$HOME`?  
 a. the absolute path of your login home directory  
 b. the relative path of your login home directory  
 c. the absolute path of the system `/home` directory  
 d. the relative path of the ROOT directory  
 e. the relative path of the system `/home` directory
141. 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. 5                    c. 4                    d. 2                    e. 3
142. Which command counts lines containing the string `refused` in only the month `October 2015` in the `denyhosts` log file?  
 a. `fgrep 'refused' denyhosts | fgrep -c '2015-10'`  
 b. `fgrep '2015-10 refused' denyhosts`  
 c. `fgrep -c 2015-10 denyhosts | fgrep refused`  
 d. `fgrep refused denyhosts | fgrep -c October 2015`  
 e. `fgrep refused denyhosts ; fgrep -c 2015-10`

143. 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."`
144. What is the link count of directory `dir` after these successful commands?
- ```
mkdir dir ; cd dir ; touch one ; mkdir two
```
- 4
  - 3
  - 2
  - 1
  - 5
145. 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:
- `/dir/bar`
  - `/bin/dir/bar/foo`
  - `/bin/dir/bar`
  - `/bin/foo/dir/bar`
  - `/foo/dir/bar`
146. 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
```
- `bash: ls: command not found`
  - no output on screen
  - `bash: cat: command not found`
  - `/bin/cat: foo: No such file or directory`
  - `bash: /bin/cat: command not found`
147. What is the resulting link count of empty directory `dir` after these successful commands? `cd dir ; touch foo ; ln foo one ; ln foo two`
- 2
  - 3
  - 1
  - 5
  - 4
148. 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
```
- `dad`
  - `mom`
  - `dad` followed by `mom`
  - `mom` followed by `dad`
  - `bash: foo: command not found`
149. 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
```
- 4
  - 5
  - 3
  - 2
  - 1
150. How many arguments are passed to the command by the shell:
- ```
echo " 1 '2 3' 4 "5 6 ' 7 "8 ' >out
```
- 5
  - 2
  - 3
  - 6
  - 4
151. Which command shows *only* names that match `rtfm`, case-insensitive?
- `echo [rR,tT,fF,mM]`
  - `echo [rRtTfFmM]`
  - `echo *[rRtTfFmM]*`
  - `echo [rR][tT][fF][mM]`
  - `echo *rtfmRTFM*`

152. How many arguments are passed to the command by the shell:
- ```
<foo foo " a 'b c' d " e ' f " g " ' >foo h
```
- 5
  - 3
  - 2
  - 4
  - 6
153. Which command moves a file into the parent directory?
- `mv file,..`
  - `mv file/..`
  - `mv .. file`
  - `mv file ..`
  - `mv ../file`
154. 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`
- `bat`
  - no output on screen
  - `/bin/ls: bat: No such file or directory`
  - `foo`
  - `bash: /bin/ls: command not found`
155. How many arguments are passed to the command by the shell:
- ```
echo 'It's "1 2" isn't it? I can't decide.
```
- 5
  - 4
  - 3
  - 6
  - 2
156. 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
```
- 4
  - 5
  - 2
  - 3
  - 1
157. How many arguments are passed to the command by the shell:
- ```
echo ' one two ' three ' four ' 5'6'
```
- 4
  - 5
  - 9
  - 1
  - 6
158. Which command line makes a file executable?
- `chmod u+x file`
  - `umask u=x file`
  - `umask -x file`
  - `umask u+x file`
  - `chmod -x file`
159. 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
```
- 2
  - 3
  - 5
  - 4
  - 1
160. Which command shows the name of the current computer:
- `hostname`
  - `history`
  - `comname`
  - `find`
  - `whois`
161. Rewrite as a simplified absolute path:
- ```
/usr/./bin/./lib/./../etc/./usr/./lib/./bin/./bar
```
- `/usr/lib/bar`
  - `/etc/bar`
  - `/usr/bar`
  - `/bar`
  - `/usr/bin/bar`



179. 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. 3                      b. 1                      c. 3 followed by 1  
d. 3 followed by 4      e. 4
180. 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. **bash: bat: command not found**  
c. **bar** followed by **foo**  
d. **foo**  
e. **foo** followed by **bar**
181. 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
```
- a. 3                      b. 1                      c. 5                      d. 2                      e. 4
182. How many files are touched? `touch '1' '2 3' '4' '5'`
- a. 1                      b. 2                      c. 5                      d. 4                      e. 3
183. If I have a directory named `/1/2`, which action would increase its *link count* by exactly one?
- a. create one file named `/1/2/3`  
b. create a directory named `/1/22`  
c. create a directory named `/1/2/3`  
d. create a directory named `/1/2`  
e. create one file named `/1/22`
184. 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
```
- a. there is a second copy of the file **bar** in file `/tmp/me`  
b. there is a second copy of the file **bar** in directory `mt`  
c. the command fails because the name `mt/../me` does not exist  
d. the directory `mt` now contains only a file named `me`  
e. the directory `mt` is now empty
185. How many arguments are passed to the command by the shell:
- ```
<bat bat -b "-a -r" >bat bat bat
```
- a. 7                      b. 5                      c. 4                      d. 6                      e. 3
186. Which command line makes pathnames `/usr/local/bin` and `/usr/bin` lead to the same directory?
- a. `ln . /usr/local`                      b. `ln -s . /usr/local`  
c. `rmdir /usr/local`                      d. `touch /usr/local`  
e. `mkdir /usr/local`
187. 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. 0                      c. 5                      d. 4                      e. 3

188. 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. 2                      b. 1                      c. 4                      d. 3                      e. 5
189. What is the output of this in an empty directory:
- ```
touch 1 13 .13 2 213 3 30 39 .31 ; echo [13]?
```
- a. **[13]?**  
b. an error message from **echo** saying **[13]?** does not exist  
c. **13**  
d. **13 30 39**  
e. **1 13 3 30 39**
190. 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. 8                      b. 0                      c. 12                      d. 5                      e. 7
191. Which file is a DOS/Windows file?
- a. ASCII text, with LF line terminators  
b. ASCII text, with CRLF line terminators  
c. ASCII text  
d. ASCII text, with no line terminators  
e. ASCII text, with CR line terminators
192. How many arguments are passed to the command by the shell:
- ```
echo 'And it's not hard, it's just logical.'
```
- a. 6                      b. 4                      c. 5                      d. 7                      e. 3
193. How many arguments are passed to the command by the shell:
- ```
<foo foo -x " " -z -r" " >foo 'foo foo'
```
- a. 6                      b. 5                      c. 8                      d. 7                      e. 9
194. 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. 4                      c. 2                      d. 5                      e. 3
195. If I have a directory named `a/b`, which action would increase its *link count* by exactly one?
- a. create a directory named `a/b2`  
b. create a file named `a/b2`  
c. create a file named `a/b/c`  
d. create a directory named `a/b/c`  
e. create a hard link to directory `b` named `b2`
196. 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
```
- a. 1                      b. 3                      c. 0                      d. 4                      e. 2

197. What is the link count of an empty directory?  
 a. 4            b. 2            c. 1            d. 0            e. 3
198. 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 `foo`  
 b. the directory `mt` is still empty  
 c. the directory `mt` now contains a file named `bar`  
 d. the directory `foo` now contains a file named `bar`  
 e. the command fails because `mt/foo` is not a directory
199. Which command line always prints just the two characters `$x` on the screen?  
 a. `echo "$x"`            b. `echo '$x'`            c. `echo $x`  
 d. `echo "$$x"`            e. `echo $$x`
200. What is the link count of directory `dir` after these successful commands?  
`mkdir dir ; cd dir ; touch foo ; mkdir a b c`  
 a. 5            b. 2            c. 1            d. 3            e. 4
201. 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`?  
 a. 123            b. 789            c. 454            d. 787            e. 456
202. 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. 3            b. 2            c. 0            d. 1            e. 4
203. Which command finds your account login userid in the password file?  
 a. `fgrep $USER /etc/passwd`  
 b. `find /etc/passwd -name $USER`  
 c. `fgrep /etc/passwd $USER`  
 d. `cat $USER /etc/passwd`  
 e. `find $USER /etc/passwd`
204. 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`

205. 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\\?`
206. 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`  
 a. `xx`  
 b. `bash: pig: command not found`  
 c. `foo`  
 d. `xx` followed by `foo`  
 e. `foo` followed by `xx`
207. What is in file `c` after this:  
`echo foo >a ; ln a b ; echo bar >>b ; ln a c ; rm a`  
 a. nothing (empty file)            b. no such file (nonexistent)  
 c. `foo`            d. `bar`  
 e. `foo` followed by `bar`
208. 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`  
 a. `foo` and `bar` are names for the same file  
 b. `foo` and `bar` are names for different files  
 c. `foo` and `bar` each have three names (six names total)  
 d. `foo` and `bar` are two of three names for the same file  
 e. this output is not possible
209. 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. 0            d. 4            e. 6
210. Which command removes *only* this four-character name containing a special character: `?xyz`?  
 a. `rm ?xyz`            b. `rm ''?xyz''`            c. `rm '?xyz'`  
 d. `rm ''?xyz`            e. `rm '?xyz'`
211. 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. `/dir/foo`            b. `/bin/dir/foo/bar`  
 c. `/bar/./dir/foo`            d. `/bin/bar/dir/foo`  
 e. `/bin/dir/foo`
212. If your terminal type is `xterm`, what is the output of this: `echo '$TERM'`  
 a. `$TERM`            b. no output on screen  
 c. `'xterm'`            d. `'$TERM'`  
 e. `xterm`

213. 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`  
 a. 2      b. 1      c. 3      d. 5      e. 4
214. 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`  
 a. 21      b. 6      c. 10      d. 2      e. 18
215. 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*`      c. `rm /a/a*`  
 d. `rm "/a/a*"`      e. `rm /a/a?`
216. How many arguments are passed to the command by the shell:  
`echo 'It's a bird! It's a plane!'`  
 a. 5      b. 4      c. 3      d. 2      e. 1
217. How many arguments are passed to the command by the shell:  
`echo " 1 2 " three ' 4 ' five"6"`  
 a. 5      b. 1      c. 9      d. 4      e. 3
218. 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. 0      c. 2      d. 4      e. 3
219. In an empty directory, what is the output on your screen after this:  
`touch 1 2 3 ; cow="*" ; echo $cow`  
 a. `*`      b. `$cow`      c. `"*"`  
 d. `1 2 3`      e. `"1 2 3"`
220. What is the link count of directory **d** after these successful commands?  
`mkdir d ; mkdir d/a ; touch d/b`  
 a. 4      b. 2      c. 5      d. 3      e. 1
221. 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. 1      b. 0      c. 4      d. 2      e. 3
222. How many files are touched? `touch 1 "2 3" ' ' 4 5`  
 a. 5      b. 7      c. 3      d. 4      e. 6

223. Which command line outputs inode/filename pairs for names in the current directory, sorted by inode number?  
 a. `ls -node * > sort -n`      b. `ls -ai | sort -n`  
 c. `ls /* | sort -node`      d. `ls -i * > sort -n`  
 e. `sort -n | ls -ai`
224. In an empty directory, what is the output on your screen after this:  
`echo one >.bar ; echo .*`  
 a. `.bar`  
 b. `. .. .bar`  
 c. an error message from `echo` saying `.*` does not exist  
 d. `one`  
 e. `.*`
225. In an empty directory, what is the output on your screen after this:  
`touch A a ; echo * ">*"`  
 a. `A a >*`      b. `A a >A a`      c. `A a`  
 d. `* >*`      e. No output
226. How many arguments are passed to the command by the shell:  
`<pig pig -x " " -z -r" " >pig pig pig`  
 a. 8      b. 6      c. 5      d. 7      e. 9
227. 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?  
 a. `rm '/a/a?'`      b. `rm /a/a?`      c. `rm "/a?"`  
 d. `rm /a/?\?`      e. `rm /a\?`
228. How many arguments are passed to the command by the shell:  
`<foo foo " a 'b c' d " e ' f " g " ' >foo`  
 a. 2      b. 3      c. 5      d. 6      e. 4
229. How many files are touched? `touch 1 "2 3" ' 4 ' 5`  
 a. 3      b. 5      c. 4      d. 6      e. 7
230. 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. **one**  
 c. **two**  
 d. **one** followed by **two**  
 e. `bash: xxx: command not found`
231. 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*`

232. How many arguments are passed to the command by the shell:

```
echo 'It's a bird! No! It's a plane!'
```

- a. 2            b. 4            c. 3            d. 1            e. 5

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

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

*This page intentionally left blank.*