

PRINT Name: _____ LAB Section:

Test Version: 877

One-Answer Multiple Choice 204 Questions – 15 of 15%

- ☞ Read **all** the words of these instructions and **both** sides (back and front) of all pages.
 - ☞ Use your full, unabbreviated name on the mark-sense form. Do not abbreviate your name.
 - ☞ Put the three-digit **Test Version** above into both **NO. OF QUESTIONS** and **NO. OF STUDENTS**
 - ☞ Fill in the bubbles with pencil only, no pen. Enter your NAME, Test Version, and answers.
 - ☞ Manage your time. Answer questions you know, first. One Answer per question.
 - ☞ The answer to the questions below about reading/doing all these test instructions is: **Jes**
-

1. **Did you read all the words of the test instructions on page one?**
 - a. **Sim** (*Yes - Portuguese*)
 - b. **Taip** (*Yes - Lithuanian*)
 - c. **Igen** (*Yes - Hungarian*)
 - d. **Tak** (*Yes - Polish*)
 - e. **Jes** (*Yes - Esperanto*)
2. **My three-digit Lab Section number is:**
 - a. My lecture section number, e.g. **010** or **020**.
 - b. My lecture room number, e.g. **T119**
 - c. My lab room number, e.g. **T303**, **T230**, **P305**, **T321**
 - d. The timetable section number of my weekly 2-hour lab period.
 - e. The Test Version number printed in the top left corner.
3. Which command line outputs inode/filename pairs for names in the current directory, sorted by inode number?
 - a. `ls -node * > sort -n`
 - b. `sort -n | ls -ai`
 - c. `ls ./* | sort -node`
 - d. `ls -i * > sort -n`
 - e. `ls -ai | sort -n`
4. 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
```

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

6. How many arguments are passed to the command by the shell:
`echo " one '2 three' 4 "five 6 ' 7 "8 ' >out`
- 5
 - 2
 - 6
 - 4
 - 3
7. File **a** contains 3 lines. File **b** contains 4 lines. How many lines are output on your screen by this command line:
`sort a | echo b`
- 3 followed by 4
 - 3
 - 1
 - 4
 - 3 followed by 1
8. What is the output on your screen of this unquoted command line:
`mkdir a ; touch b a/b1 a/b2 ; find a -name b*`
- `b a/b1 a/b2`
 - `a/b1 a/b2`
 - no output
 - `b1 b2`
 - `b`
9. In an empty directory, how many words are in file **a** after this:
`echo It's redirected >b isn't it\? ; ls >a`
- 3
 - 2
 - 4
 - 0
 - 1
10. If `/bin/foo` is a program that outputs `hi` and `/usr/bin/foo` is a program that outputs `mom` what would be the output on your screen of this two command sequence: `PATH=/etc:/usr/bin:/bin ; foo`
- `hi` followed by `mom`
 - `mom` followed by `hi`
 - `bash: foo: command not found`
 - `hi`
 - `mom`
11. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this command line:
`ln a e ; ln b d ; ln d c ; cp d e ; sort a b e d >c`
- 4
 - 7
 - 10
 - 6
 - 12
12. 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/??`
 - `rm dir/?1`
 - `rm dir/*1`
 - `rm dir/1*`
 - `rm dir/\??`

13. In an empty directory, what is in file `count` after this command line: `ls ??? | wc -w >count`
- 1
 - 1 1 2
 - 1 1 1
 - nothing (empty file)
 - 0
14. What is the output on your screen after this two-command sequence if run in a directory containing 8 files with names that are all the numbers from 1 to 8 inclusive: `cow="*" ; echo "$cow"`
- `$cow`
 - the file names 1 through 8
 - `'$cow'`
 - *
 - the file names 1 through 8, surrounded by quotes
15. If file `foo` occupies one disk block, how many disk blocks are in use after this sequence of commands:
`cp foo bar ; ln bar one ; cp one two ; cp one xxx`
- 3
 - 2
 - 1
 - 5
 - 4
16. What displays on your screen given this command:
`date >date ; pwd >pwd ; head date | tail pwd`
- `head` displays the `date` and `tail` displays the `pwd`
 - only the `date` displays because `tail` ignores the pipe
 - `tail` reads the pipe and the `pwd` and displays both together
 - nothing displays because `tail` ignores the pipe
 - only the `pwd` displays because `tail` ignores the pipe
17. What is the output on your screen after this two-command sequence if run in a directory containing 8 files with names that are all the numbers from 1 to 8 inclusive: `cow="*" ; echo '$cow'`
- the file names 1 through 8, surrounded by quotes
 - `'$cow'`
 - `$cow`
 - *
 - the file names 1 through 8
18. How many arguments are passed to the command by the shell: `<bat bat -b "-a -r" >bat bat bat`
- 3
 - 4
 - 5
 - 6
 - 7
19. How many arguments are passed to the command by the shell:
`echo 'And it's not hard, it's just logical.'`
- 3
 - 5
 - 4
 - 7
 - 6

20. If your terminal type is `xterm`, what is the output of this command line? `echo '$TERM'`
- a. no output on screen
 - b. `'xterm'`
 - c. `'$TERM'`
 - d. `$TERM`
 - e. `xterm`
21. How many files are touched or created? `touch 1 "2 3" '4' 5`
- a. 6
 - b. 2
 - c. 4
 - d. 3
 - e. 5
22. 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` each have three names (six names total)
  - c. `foo` and `bar` are names for different files
  - d. `foo` and `bar` are two of three names for the same file
  - e. `foo` and `bar` are names for the same file
23. To change to the parent directory, do this:
- a. `pwd`
  - b. `cd .`
  - c. `pwd ..`
  - d. `cd ..`
  - e. `cd`
24. 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. 4
 - b. 1
 - c. 5
 - d. 2
 - e. 3
25. How many arguments are passed to the command by the shell:
- ```
<foo foo -x " " -z -r" " >foo 'foo foo'
```
- a. 6
  - b. 5
  - c. 9
  - d. 7
  - e. 8
26. Which one of these names is usually a shell environment variable?
- a. `FOOBAR`
  - b. `fooBar`
  - c. `foobar`
  - d. `FooBar`
  - e. `FooBar`

27. 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` are two of three names for this file
 - this output is not possible
 - `foo` and `bar` each have two names (four names total)
 - `foo` and `bar` are names for the same file
 - `foo` and `bar` each have three names (six names total)
28. What is in file `c` after this command line:
- ```
echo B >b ; ln b a ; echo A >a ; ln a c ; rm a b
```
- `A`
  - nothing (empty file)
  - `B`
  - no such file (nonexistent)
  - `A` followed by `B`
29. If `/bin/xxx` is a program that outputs `one` and `/usr/bin/xxx` is a program that outputs `two`, what would be the output on your screen of this two command sequence: `PATH=/etc:/usr/bin:/usr:/bin ; /bin/xxx`
- `two` followed by `one`
  - `bash: /bin/xxx: command not found`
  - `one` followed by `two`
  - `one`
  - `two`
30. 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`
  - `/foo/dir/bar`
  - `/bin/foo/dir/bar`
31. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are output on your screen by this command line:
- ```
cat b | cat a
```
- 2 followed by 3
 - 3 followed by 2
 - 2
 - 5
 - 3
32. What is the link count of directory `d` after these successful commands?
- ```
mkdir d ; mkdir d/a d/b ; touch d/c d/e
```
- 5
  - 2
  - 1
  - 4
  - 3
33. What displays on your screen given this command: `ls >ls ; wc ls >wc ; sort ls | cat wc`
- `cat` reads the pipe and the `wc` and displays both together
  - `sort` displays the `ls` and `cat` displays the `wc`
  - only the `wc` displays because `cat` ignores the pipe
  - only the `ls` displays because `cat` ignores the pipe
  - nothing displays because `cat` ignores the pipe

34. What is in file **c** after this command line:  
`echo foo >a ; ln a b ; echo bar >>b ; ln a c ; rm a`
- a. **foo** followed by **bar**
  - b. **bar**
  - c. no such file (nonexistent)
  - d. nothing (empty file)
  - e. **foo**
35. What is the output on your screen after this command line: `mkdir foo ; rmdir foo | wc -c`
- a. no output
  - b. **3**
  - c. **0**
  - d. **4**
  - e. **1**
36. What is the output on your screen after this command line:  
`echo 1 >x ; ln x y ; echo 2 >>y ; sort x`
- a. **2** followed by **1**
  - b. **1** followed by **2**
  - c. no output
  - d. **2**
  - e. **1**
37. If file **one** occupies one disk block, how many disk blocks are in use after this sequence of commands:  
`cp one foo ; ln foo two ; ln two bar ; ln one cow`
- a. **5**
  - b. **3**
  - c. **1**
  - d. **4**
  - e. **2**
38. 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. **3**
  - b. **4**
  - c. **5**
  - d. **2**
  - e. **1**
39. 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`
- a. **5**
  - b. **3**
  - c. **2**
  - d. **1**
  - e. **4**
40. How many files are touched or created? `touch '1 "2 3 '4" ' 5`
- a. **4**
  - b. **3**
  - c. **1**
  - d. **2**
  - e. **5**

41. How many arguments are passed to the command by the shell:  
`<wc wc " 1 '2 3' 4 " 5 6 ' 7 " 8 " ' >wc 9`
- 5
  - 2
  - 3
  - 6
  - 4
42. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this command line:  
`ln a e ; ln b d ; ln d c ; cat e b >c`
- 5
  - 2
  - 3
  - 4
  - 0
43. 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 size of this directory.
  - The number 128 is the inode number of this directory.
  - The number 32 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.
44. What is the link count of directory **dir** after these successful commands?  
`mkdir dir ; mkdir dir/foo ; touch dir/bar`
- 4
  - 5
  - 1
  - 3
  - 2
45. 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`
46. What is the link count of directory **z** after these successful commands?  
`mkdir z ; cd z ; touch a b ; mkdir c d e`
- 6
  - 4
  - 7
  - 3
  - 5
47. What is in file **out** after this command line:  
`echo me >a ; ln a b ; echo hi >b ; ln a out ; rm a b`
- no such file (nonexistent)
  - me**
  - me** followed by **hi**
  - nothing (empty file)
  - hi**

48. 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
  - 1
  - 3
  - 5
  - 2
49. 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`
- 4
  - 1
  - 3
  - 2
  - 5
50. 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`
- 2
  - 3
  - 5
  - 4
  - 6
51. What is the output on your screen after this command line:  
`echo one >x ; ln x y ; echo two >>y ; sort x`
- two
  - one followed by two
  - one
  - two followed by one
  - no output
52. What is true about this output from `ls -il foo bar`  

```
15 -r-x-----x 2 me me 3 Jan 1 1:00 foo
15 -r-x-----x 2 me me 3 Jan 1 1:00 bar
```
- this output is not possible
  - foo** and **bar** are two of three names for this file
  - foo** and **bar** are names for the same file
  - foo** and **bar** each have three names (six names total)
  - foo** and **bar** are names for different files
53. Which command removes *only* this four-character name containing a special character: **\*xyz**
- `rm "*xyz"`
  - `rm *xyz`
  - `rm "*"xyz"`
  - `rm ''*xyz''`
  - `rm '*xyz'`



54. What is in file **c** after this command line:  
`echo A >a ; ln a b ; echo B >b ; ln a c ; rm a b`
- B**
  - A** followed by **B**
  - nothing (empty file)
  - A**
  - no such file (nonexistent)
55. What is the link count of directory **d** after these successful commands?  
`mkdir d ; cd d ; touch f ; ln f a ; ln f b`
- 3
  - 4
  - 5
  - 2
  - 1
56. Create a symbolic link under **/usr** named **bar** that has target **xy**:
- `ln -s /usr/bar 'xy'`
  - `ln -s 'xy' /bar/usr`
  - `ln -s '/usr/xy' /usr/bar`
  - `ln -s /usr/bar '/usr/xy'`
  - `ln -s 'xy' '/usr/bar'`
57. In an empty directory, how many files are created by this command line: `touch 1 "2 3" '4' 5`
- 7
  - 3
  - 4
  - 6
  - 5
58. 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:
- `/bar/./dir/foo`
  - `/dir/foo`
  - `/bin/dir/foo/bar`
  - `/bin/bar/dir/foo`
  - `/bin/dir/foo`
59. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this command line:  
`ln a d ; ln d c ; cp c b ; sort a b d >c`
- 6
  - 0
  - 4
  - 5
  - 2
60. Rewrite as a simplified absolute path:  
`/usr/./bin/./lib/./../etc/./usr/./lib/./bin/./bar`
- `/usr/bar`
  - `/etc/bar`
  - `/usr/bin/bar`
  - `/usr/lib/bar`
  - `/bar`

61. In an empty directory, what is the output on your screen after this command line:  
`echo one >.bar ; echo .*`
- a. `one`
  - b. `.bar`
  - c. an error message from `echo` saying `.*` does not exist
  - d. `. .. .bar`
  - e. `.*`
62. What does *quoting* mean on a shell command line?
- a. using a leading tilde ("`~`") on a pathname to mean your HOME directory
  - b. using more than one pathname argument to a command, e.g. `rm a b c`
  - c. setting the `PS1` variable to be your shell prompt
  - d. typing a "control" character using the [CTRL] key
  - e. turning off the special meaning of shell meta-characters
63. What is the link count of directory `dir` after these successful commands?  
`mkdir dir ; cd dir ; touch one ; mkdir two`
- a. 1
  - b. 3
  - c. 5
  - d. 4
  - e. 2
64. 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. 2
  - b. no output
  - c. 1
  - d. 1 followed by 2
  - e. 2 followed by 1
65. 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]'`
66. If `/bin/pig` is a program that outputs `xx` and `/usr/bin/pig` is a program that outputs `foo` what would be the output on your screen of this two command sequence: `PATH=/home:/bin:/dev:/usr/bin ; pig`
- a. `foo`
  - b. `xx`
  - c. `bash: pig: command not found`
  - d. `foo` followed by `xx`
  - e. `xx` followed by `foo`

67. 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
```
- 2
 - 3
 - 0
 - 4
 - 1
68. What command will recursively show disk usage in directories?
- ls
 - tree
 - du
 - find
 - df
69. 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/*
 - rm /a*
70. 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:
- /bar/foo
 - /foo
 - /etc/foo
 - /etc/bar/foo
 - /etc/foo/bar
71. What is the link count of directory `z` after these successful commands?
- ```
mkdir z ; cd z ; touch a ; ln a b ; ln a c
```
- 2
  - 1
  - 4
  - 5
  - 3
72. In an empty directory, what is the output on your screen after this command line: `touch a ; ls | wc -l`
- 2
  - no output
  - 0
  - 1
  - 3
73. How many arguments are passed to the command by the shell: `<cow cow "-x" -y '-z' >cow cow`
- 2
  - 5
  - 4
  - 3
  - 6

74. 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`  
 a. 1  
 b. 5  
 c. 3  
 d. 2  
 e. 4
75. 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 names for different files  
 b. **foo** and **bar** are two of three names for the same file  
 c. this output is not possible  
 d. **foo** and **bar** are names for the same file  
 e. **foo** and **bar** each have three names (six names total)
76. In an empty directory, what is the output on your screen after this command line:  
`echo hi >a ; ls | wc -w`  
 a. 0  
 b. 2  
 c. 1  
 d. no output  
 e. a
77. In an empty directory, what is the output on your screen after this command line:  
`echo hi >a ; mv a b ; ln b c ; ls >wc -l`  
 a. no output  
 b. a  
 c. 0  
 d. 1  
 e. 2
78. 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`  
 a. 1  
 b. 3  
 c. 4  
 d. 2  
 e. 0
79. Which command line shows the current date?  
 a. `date | bash`  
 b. `bash >date ; cat date`  
 c. `bash date`  
 d. `bash <date`  
 e. `echo date | bash`

80. If `/bin/xxx` is a program that outputs **one** and `/usr/bin/xxx` is a program that outputs **two**, what would be the output on your screen of this two command sequence:
- ```
PATH=/bin/xxx:/usr/bin/xxx:/etc/passwd ; xxx
```
- one** followed by **two**
 - two**
 - bash: xxx: command not found**
 - one**
 - two** followed by **one**
81. In an empty directory, what is the output on your screen of this command line:
- ```
echo hi >foo ; cp foo bar | wc -l
```
- 2**
  - 1**
  - 0**
  - 3**
  - no output
82. How many arguments are passed to the command by the shell:
- ```
echo 'It's a bird! No! It's a plane!'
```
- 1**
 - 2**
 - 4**
 - 3**
 - 5**
83. If I have a directory named **a/b**, which action would increase its *link count* by exactly one?
- create a file named **a/b/c**
 - create a directory named **a/b2**
 - create a hard link to directory **b** named **b2**
 - create a file named **a/b2**
 - create a directory named **a/b/c**
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-- 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
```
- 3**
  - 0**
  - 2**
  - 4**
  - 1**
85. 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/a?**
  - rm '/a/a?'**
  - rm /a/?\?**
  - rm /a\?**

86. How many arguments are passed to the command by the shell:  
`echo ' one two ' three ' four ' 5'6'`
- a. 1
  - b. 5
  - c. 6
  - d. 9
  - e. 4
87. How many arguments are passed to the command by the shell: `echo " 1 2 " three ' 4 ' five"6"`
- a. 4
  - b. 1
  - c. 5
  - d. 9
  - e. 3
88. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this command line:  
`ln a d ; ln d e ; ln b f >c`
- a. 2
  - b. 5
  - c. 3
  - d. 0
  - e. 4
89. Dereference the following symlink **xyz** into its equivalent absolute path:  
`ln -s ../../a/./b/./bar /tmp/a/b/xyz`
- a. /tmp/a/b/bar
  - b. /tmp/b/xyz
  - c. /tmp/a/bar
  - d. /tmp/bar
  - e. /tmp/b/bar
90. How many arguments are passed to the command by the shell:  
`<foo foo " a 'b c' d " e ' f " g " ' >foo h`
- a. 3
  - b. 5
  - c. 2
  - d. 6
  - e. 4
91. What is the link count of directory **d** after these successful commands?  
`mkdir d ; cd d ; touch a ; mkdir b c`
- a. 2
  - b. 5
  - c. 3
  - d. 4
  - e. 6
92. The output of the **whoami** command is:
- a. your userid
  - b. a list of accounts in the password file
  - c. a list of users logged in to the system
  - d. the current directory
  - e. your HOME directory

93. If your **PATH** variable contains **/bin:/usr/bin**, what is the output of this command line: **echo '\$PATH'**
- '\$PATH'**
  - '/bin:/usr/bin'**
  - echo: \$PATH: No such file or directory**
  - /bin:/usr/bin**
  - \$PATH**
94. What is the output on your screen after this command line: **echo hi >a ; cp a b | wc -c**
- 3**
  - no output**
  - 0**
  - 2**
  - 1**
95. What is usually in the environment variable **\$HOME**?
- the absolute path of your login home directory
  - the relative path of your login home directory
  - the relative path of the system **/home** directory
  - the relative path of the **ROOT** directory
  - the absolute path of the system **/home** directory
96. Which option to **ls** displays the directory itself and not its contents?
- R**
  - d**
  - a**
  - i**
  - l**
97. 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
```
- 1**
 - 4**
 - 3**
 - 2**
 - 0**
98. If **/bin/foo** is a program that outputs **one** and **/usr/bin/foo** is a program that outputs **two**, what would be the output on your screen of this two command sequence: **PATH=/etc:/usr/bin:/usr:/bin:/dev ; foo**
- one**
 - one** followed by **two**
 - two**
 - bash: foo: command not found**
 - two** followed by **one**
99. How many arguments are passed to the command by the shell: **<cow cow "-x "-y '-z' >cow cow**
- 3**
 - 6**
 - 4**
 - 7**
 - 5**

100. How do you execute the program `foo` in the current directory?
- a. `$HOME/foo`
 - b. `./foo`
 - c. `/foo`
 - d. `foo/`
 - e. `foo/.`
101. What is the link count of directory `d` after these successful commands?
- ```
mkdir d ; touch f ; cd d ; ln ../f x
```
- a. 2
  - b. 4
  - c. 3
  - d. 5
  - e. 1
102. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `c` after this command line:
- ```
ln a d ; ln d c ; cat a b >c
```
- a. 5
 - b. 0
 - c. 2
 - d. 4
 - e. 3
103. 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
```
- a. 1
  - b. 3
  - c. 5
  - d. 2
  - e. 4
104. Which of the following is true, given this long directory listing:
- ```
drwxr-x--x 128 me me 32 Jan 1 1:00 dir
```
- a. The number 128 is the inode number of this directory.
 - b. The number 128 is the size of this directory.
 - c. The number 32 is the inode number of this directory.
 - d. The number 128 is the count of links (names) this directory has.
 - e. The number 32 is the count of links (names) this directory has.
105. What is the output on your screen after this two command sequence:
- ```
PATH=/bin/cat:/bin/sh:/bin/ls ; ls nosuchfile
```
- a. `bash: /bin/ls: command not found`
  - b. `bash: ls: command not found`
  - c. `bash: /bin/sh: No such file or directory`
  - d. `ls: /bin/ls: command not found`
  - e. `ls: nosuchfile: No such file or directory`



106. 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. 2
b. 3
c. 4
d. 1
e. 0
107. What is the output on your screen after this command line: `echo hi >out | wc -l`
- a. 0
b. 2
c. 3
d. no output
e. 1
108. What is the link count of directory `dir` after these successful commands?
- ```
mkdir dir ; cd dir ; touch foo ; mkdir a b c
```
- a. 3  
b. 4  
c. 2  
d. 1  
e. 5
109. 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`
110. Which command line would show the index (inode) number of a file?
- a. `cat -i file`  
b. `ls -l file`  
c. `cat -l file`  
d. `ls -i file`  
e. `find -i file`
111. 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\*`
112. If the file `bat` contained the word `foo`, what would be the output on your screen of this two command sequence:
- ```
PATH=/bin/cat:/bin/who:/bin/ls ; cat bat
```
- a. `bash: cat: command not found`
b. no output on screen
c. `bat`
d. `cat: bat: No such file or directory`
e. `foo`

113. If I have a directory named `/1/2`, which action would increase its *link count* by exactly one?
- create one file named `/1/22`
 - create a directory named `/1/22`
 - create a directory named `/1/2`
 - create one file named `/1/2/3`
 - create a directory named `/1/2/3`
114. How many arguments are passed to the command by the shell:
- ```
echo "cow "y " bat 'man x' " pig'a "hop' a b
```
- 4
  - 5
  - 11
  - 6
  - 7
115. 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
```
- 4
 - 0
 - 1
 - 3
 - 2
116. In an empty directory, how many files are created by this command line: `touch a "b c" ' ' d e`
- 6
 - 4
 - 5
 - 7
 - 3
117. Which command shows the name of the current computer:
- `comname`
 - `whoami`
 - `find`
 - `hostname`
 - `history`
118. If file `foo` occupies one disk block, how many disk blocks are in use after this sequence of commands:
- ```
cp foo bar ; ln bar one ; cp one two ; ln one pig
```
- 1
  - 3
  - 5
  - 4
  - 2
119. How many files are touched or created? `touch '1 "2 3 '4'" '5`
- 4
  - 1
  - 3
  - 2
  - 5

120. If `/bin/xxx` is a program that outputs **one** and `/usr/bin/xxx` is a program that outputs **two**, what would be the output on your screen of this two command sequence: `PATH=/usr:/usr/bin:/etc:/bin ; xxx`
- one**
  - one** followed by **two**
  - two**
  - two** followed by **one**
  - bash: xxx: command not found**
121. 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/bar/foo`
  - `/bar/foo`
  - `/etc/foo`
  - `/foo`
  - `/etc/foo/bar`
122. 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
```
- 4
 - 6
 - 2
 - 5
 - 3
123. 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
```
- 2
  - 3
  - 4
  - 1
  - 0
124. 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
```
- 3
 - 0
 - 1
 - 4
 - 2
125. If the file `bat` contained the word `foo`, what would be the output on your screen of this two command sequence:
- ```
PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/ls bat
```
- foo**
  - bat**
  - `/bin/ls: bat: No such file or directory`
  - no output on screen
  - bash: /bin/ls: command not found**

126. What is the link count of directory **foo** after these successful commands?  
`mkdir foo ; cd foo ; touch a b c`
- a. 4
  - b. 1
  - c. 3
  - d. 2
  - e. 5
127. How many arguments are passed to the command by the shell:  
`<foo foo " a 'b c' d " e ' f " g " ' >foo`
- a. 5
  - b. 3
  - c. 2
  - d. 4
  - e. 6
128. Which of the following **PATH** statements makes the most sense?
- a. `PATH=/bin:/etc/passwd:/usr/bin`
  - b. `PATH=/bin:/usr/bin:/etc/passwd`
  - c. `PATH=/bin/bash:/usr/bin:/bin`
  - d. `PATH=/bin/ls:/etc/passwd:/usr/bin`
  - e. `PATH=/bin:/usr/bin`
129. If `/bin/foo` is a program that outputs **mom** and `/usr/bin/foo` is a program that outputs **dad**, what would be the output on your screen of this two command sequence:  
`PATH=/dev:/usr/bin:/usr:/bin:/etc ; /bin/foo`
- a. **dad** followed by **mom**
  - b. **mom** followed by **dad**
  - c. **mom**
  - d. **bash: /bin/foo: command not found**
  - e. **dad**
130. What is the link count of directory **dir** after these successful commands?  
`mkdir dir ; cd dir ; touch a b c ; mkdir d e`
- a. 4
  - b. 5
  - c. 7
  - d. 3
  - e. 2
131. 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. 3
  - b. 2
  - c. 4
  - d. 6
  - e. 5
132. How many arguments are passed to the command by the shell:  
`<bar bar -b "-a" '-r' >bar bar bar`
- a. 3
  - b. 4
  - c. 5
  - d. 6
  - e. 7

133. 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. no output
  - b. **two** followed by **one**
  - c. **one**
  - d. **one** followed by **two**
  - e. **two**
134. What is the link count of directory **dir** after these successful commands?  
`mkdir dir ; touch foo ; cd dir ; ln ../foo bar`
- a. 5
  - b. 2
  - c. 1
  - d. 3
  - e. 4
135. In an empty directory, what is in file **out** after this command line: `ls nosuchfile | wc -l >out`
- a. nothing (empty file)
  - b. 0
  - c. 1
  - d. **out**
  - e. **nosuchfile**
136. 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** followed by **two** and **ten**
  - b. **ten**
  - c. **one**
  - d. **two**
  - e. no output on screen
137. How many arguments are passed to the command by the shell:  
`echo " 1 '2 3' 4 "5 6 ' 7 "8 ' >out`
- a. 2
  - b. 3
  - c. 6
  - d. 4
  - e. 5
138. What is the link count of directory **d** after these successful commands?  
`mkdir d ; mkdir d/a ; touch d/b`
- a. 3
  - b. 4
  - c. 2
  - d. 1
  - e. 5
139. Which command usually goes in your **.bash\_profile** file?
- a. **.bash\_profile source**
  - b. **cat .bashrc**
  - c. **source ./bashrc**
  - d. **source ./bash\_profile**
  - e. **.bashrc source**

140. 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?`
  - `rm /a/a[*]`
  - `rm /a/a?`
141. 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
 - 4
 - 3
 - 1
 - 5
142. What is the output on your screen after this two-command sequence if run in a directory containing 8 files with names that are all the numbers from 1 to 8 inclusive: `cow="*" ; echo "$cow"`
- the file names 1 through 8
 - *
 - `$cow`
 - the file names 1 through 8, surrounded by quotes
 - '`$cow`'
143. How many arguments are passed to the command by the shell:
- ```
<f z " a 'b c' d " 1 2 ' g " h " ' >z
```
- 6
  - 5
  - 2
  - 3
  - 4
144. 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
 - 5
 - 3
 - 4
 - 1
145. 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
```
- this output is not possible
  - `foo` and `bar` are names for different files
  - `foo` and `bar` each have three names (six names total)
  - `foo` and `bar` are two of three names for the same file
  - `foo` and `bar` are names for the same file

146. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this command line:  
`ln a d ; ln b e ; cp d e >c`
- a. 3
  - b. 5
  - c. 2
  - d. 0
  - e. 4
147. How many arguments are passed to the command by the shell: `echo " 1 2 "three ' 4 ' five"6"`
- a. 1
  - b. 5
  - c. 4
  - d. 3
  - e. 9
148. 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. 5
  - b. 1
  - c. 3
  - d. 2
  - e. 4
149. Which command line below shows only lines 6-10 of file **foo**?
- a. `head -10 foo | tail -5`
  - b. `head -6 foo | tail -10`
  - c. `tail -15 foo | head -5`
  - d. `tail -10 foo | head -6`
  - e. `head -10 foo | tail -6`
150. What is the link count of an empty directory?
- a. 3
  - b. 0
  - c. 1
  - d. 2
  - e. 4
151. 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:
- a. **/bar**
  - b. **/bin/bar**
  - c. **/bin/foo/bar**
  - d. **/bin/bar/foo**
  - e. **/foo/bar**
152. 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'`

153. 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. 5
  - c. 1
  - d. 4
  - e. 2
154. If `/bin/bat` is a program that outputs **foo** and `/usr/bin/bat` is a program that outputs **hi** what would be the output on your screen of this two command sequence: `PATH=/usr:/usr/bin:/bin ; bat`
- a. **foo**
  - b. **hi** followed by **foo**
  - c. **foo** followed by **hi**
  - d. **hi**
  - e. **bash: bat: command not found**
155. The option to `ls` that shows inode (index) numbers is:
- a. `-l`
  - b. `-i`
  - c. `-x`
  - d. `-1`
  - e. `-a`
156. Which command line makes pathnames `/usr/local/bin` and `/usr/bin` lead to the same directory?
- a. `ln -s . /usr/local`
  - b. `rmdir /usr/local`
  - c. `touch /usr/local`
  - d. `mkdir /usr/local`
  - e. `ln . /usr/local`
157. If the file `pig` contained the word **bar**, what would be the output on your screen of this two command sequence:  
`PATH=/etc/passwd:/bin/ls:/bin/who ; /bin/cat pig`
- a. **bar**
  - b. **pig**
  - c. **/bin/cat: pig: No such file or directory**
  - d. no output on screen
  - e. **bash: /bin/cat: command not found**
158. How many arguments are passed to the command by the shell:  
`<bar bar -b"-a '-r' >bar" bar >out`
- a. 6
  - b. 3
  - c. 2
  - d. 4
  - e. 5
159. What is usually in the environment variable `$PATH`?
- a. the absolute path of the system `/path` directory
  - b. a colon-separated list of your `passwd` file fields
  - c. a colon-separated list of directories containing command names
  - d. the absolute path of your login home directory
  - e. the absolute path of your login shell



160. You enter this `cp a/b c/`  
and get `cp: a: No such file or directory`  
because:
- directory `c` does not exist
  - directory `a` does not exist
  - you forgot to specify the destination file name after `c/`
  - pathname `a` exists but is a file, not a directory
  - the command `cp` is not in your search PATH
161. 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
```
- 2
 - 1
 - 0
 - 4
 - 3
162. Which command line always prints just the two characters `$x` on the screen?
- `echo '$x'`
 - `echo "$x"`
 - `echo $$x`
 - `echo "$$x"`
 - `echo $x`
163. 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
```
- 4
  - 3
  - 1
  - 2
  - 0
164. 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
```
- 4
 - 0
 - 2
 - 3
 - 1
165. What is the output of this command line if run in an empty directory: `touch A a ; echo * >"*" ; ls`
- `* A a`
 - `A a >A a`
 - `* >*`
 - `A a >*`
 - No output

166. If `/bin/pig` is a program that outputs `hi` and `/usr/bin/pig` is a program that outputs `foo` what would be the output on your screen of this two command sequence: `PATH=/etc:/usr/bin:/bin ; pig`
- `foo`
 - `foo` followed by `hi`
 - `bash: pig: command not found`
 - `hi`
 - `hi` followed by `foo`
167. What is in file `foo` after this command line:
`echo hi >a ; ln a b ; echo me >b ; ln a foo ; rm a b`
- nothing (empty file)
 - `me`
 - `hi`
 - no such file (nonexistent)
 - `hi` followed by `me`
168. Which of these statements is true?
- If `/x` is an empty directory, `sort /x/*` produces an error message.
 - Only backslashes are strong enough to stop GLOB patterns from expanding.
 - Only single quotes are strong enough to stop GLOB patterns from expanding.
 - If `/y` is an empty directory, `echo /y/*` produces an error message.
 - Only double quotes are strong enough to stop GLOB patterns from expanding.
169. If `/bin/foo` is a program that outputs `mom` and `/usr/bin/foo` is a program that outputs `dad` what would be the output on your screen of this two command sequence: `PATH=/bin/foo:/usr/bin/foo:/usr ; foo`
- `mom`
 - `mom` followed by `dad`
 - `bash: foo: command not found`
 - `dad`
 - `dad` followed by `mom`
170. If `foo` were a readable empty file, what would be the output on your screen of this two command sequence:
`PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/cat foo`
- `bash: ls: command not found`
 - `/bin/cat: foo: No such file or directory`
 - `bash: /bin/cat: command not found`
 - `bash: cat: command not found`
 - no output on screen
171. What is the output of this command line if run in an empty directory: `touch A a ; echo * ">*"`
- `* >*`
 - `A a >*`
 - `A a`
 - No output
 - `A a >A a`
172. 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`
- 4
 - 3
 - 2
 - 0
 - 1

173. How many arguments are passed to the command by the shell:
`echo 'It's "1 2" isn't it? I can't decide.`
 a. 2
 b. 5
 c. 6
 d. 3
 e. 4
174. How many arguments are passed to the command by the shell:
`<foo foo " a 'b c' d " e f ' g " h " ' >foo`
 a. 4
 b. 2
 c. 5
 d. 3
 e. 6
175. What is usually in the environment variable `$SHELL`?
 a. the absolute path of the system `/shell` directory
 b. the relative path of the system `/shell` directory
 c. the absolute path of your login shell
 d. the relative path of your login shell
 e. the relative path of the `/home/shell` directory
176. How many files are touched or created? `touch 1 "2 3 ' 4 '" 5`
 a. 4
 b. 3
 c. 5
 d. 1
 e. 2
177. How many arguments are passed to the command by the shell: `echo 'It's a bird! It's a plane!'`
 a. 4
 b. 1
 c. 2
 d. 5
 e. 3
178. 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. 4
 b. 0
 c. 1
 d. 3
 e. 2
179. If `/bin/foo` is a program that outputs `dad` and `/usr/bin/foo` is a program that outputs `mom` what would be the output on your screen of this two command sequence: `PATH=/usr:/etc:/bin:/usr/bin ; foo`
 a. `dad`
 b. `dad` followed by `mom`
 c. `bash: foo: command not found`
 d. `mom` followed by `dad`
 e. `mom`

180. If `/bin/foo` is a program that outputs **one** and `/usr/bin/foo` is a program that outputs **two**, what would be the output on your screen of this two command sequence:
`PATH=/dev:/usr/bin:/usr:/bin:/etc ; /bin/foo`
- `bash: /bin/foo: command not found`
 - one** followed by **two**
 - two**
 - two** followed by **one**
 - one**
181. What is in the local variable `$$` ?
- the first argument of the previous command line
 - the process ID of the current shell
 - the command name of the previous command line
 - `$$` is not a valid variable name
 - the cpu cost of the current session, in dollars
182. Rewrite as a simplified absolute path: `../../var/./a/../../var/b/../../etc/./bar/./foo`
- `/var/b/foo`
 - `/var/a/foo`
 - `/etc/foo`
 - `/var/foo`
 - `/etc/bar/foo`
183. 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
 - 5
 - 2
 - 1
 - 4
184. 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`
- 6
 - 4
 - 2
 - 3
 - 5
185. A "dangling symlink" is a symlink to:
- a non-existent target
 - a parent directory
 - a special device file
 - a directory
 - the current directory
186. What is the link count of directory **d** after these successful commands?
`mkdir d ; mkdir d/a ; mkdir d/b ; mkdir d/b/c`
- 4
 - 1
 - 2
 - 3
 - 5

187. 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
```
- 2
  - 0
  - 4
  - 3
  - 1
188. If `/bin/foo` is a program that outputs **one** and `/usr/bin/foo` is a program that outputs **two**, what would be the output on your screen of this two command sequence: `PATH=/bin/ls:/home:/usr/bin/cat:/etc ; foo`
- one** followed by **two**
  - bash: foo: command not found**
  - two**
  - one**
  - two** followed by **one**
189. 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/*`
  - `rm /a*`
190. 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=/bin:/usr/bin:.`
  - `$PATH=./$HOME:/usr/bin`
  - `PATH=./$HOME:/usr/bin`
  - `PATH=/usr/bin/./$HOME`
  - `$PATH=/usr/bin:./bin`
191. 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
```
- 1
 - 3
 - 5
 - 4
 - 2
192. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `e` after this command line:
- ```
ln a d ; cp a f ; ln d c ; ln c e ; cat a b d f >e
```
- 5
  - 9
  - 6
  - 2
  - 3

193. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **d** (not in **c**) after this command line:

```
ln a d ; ln d c ; ln c e ; cat a a b b c c d d e e >c
```

- a. 18
- b. 2
- c. 6
- d. 10
- e. 21

194. If **/bin/bat** is a program that outputs **foo** and **/usr/bin/bat** is a program that outputs **bar** what would be the output on your screen of this two command sequence: **PATH=/usr:/usr/bin:/bin ; bat**

- a. **bar**
- b. **foo**
- c. **foo** followed by **bar**
- d. **bash: bat: command not found**
- e. **bar** followed by **foo**

195. If **/bin/prg** is a program that outputs **hi** and **/usr/bin/prg** is a program that outputs **foo** what would be the output on your screen of this two command sequence: **PATH=/etc:/usr/bin:/bin ; prg**

- a. **hi** followed by **foo**
- b. **bash: prg: command not found**
- c. **hi**
- d. **foo** followed by **hi**
- e. **foo**

196. 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. 456
- b. 787
- c. 789
- d. 454
- e. 123

197. If your **PATH** contained only the file names **/bin/sh**, **/bin/cat**, and **/bin/ls**, then what would be the output on your screen of this command: **cat /etc/passwd**

- a. **bash: /bin/sh: command not found**
- b. **bash: /bin/cat: no such file or directory**
- c. **bash: cat: command not found**
- d. **cat: /etc/passwd: command not found**
- e. **cat: bash: no such file or directory**

198. 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. define the alias in my file **\$HOME/.bashrc**
- d. put the alias into the **/etc/passwd** file for next log in
- e. put the alias into the **grub.conf** file for next log in

199. 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**
- b. **ten**
- c. **one** followed by **ten** and **two**
- d. **two**
- e. no output on screen

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

```
< pig pig -x " " -z -r" " > pig pig pig
```

- a. 6
- b. 5
- c. 9
- d. 7
- e. 8

201. 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
```

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

202. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this command line:

```
sort a b >c ; cat a >>b ; cat c b >c a
```

- a. 5
- b. 0
- c. 12
- d. 7
- e. 8

203. How many files are touched or created? `touch 1 "2 3" ' ' 4 5`

- a. 5
- b. 6
- c. 4
- d. 3
- e. 7

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

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