PRINT Name: _____ **LAB Section:** ☐☐☐

Test Version: 721   **One-Answer Multiple Choice**   **241 Questions – 25 of 25%**

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

2. **My three-digit Lab Section number is:**
   a. My lab room number, e.g. **B384**.
   b. My lecture section number **400** or **410**.
   c. My lecture room number, e.g. **T130**.
   d. The section number of my weekly 2-hour lab period.
   e. The Test Version number printed in the top left corner.

3. If I have a directory named **c/d**, which action would increase its *link count* by exactly one?
   a. create a directory named **c/d2**
   b. create a file named **c/d2**
   c. create a file named **c/d/e**
   d. create a hard link to directory **d** named **d2**
   e. create a directory named **c/d/e**

4. What displays on your screen given this command:
   ls >ls ; wc ls >wc ; sort ls | cat wc
   a. only the **ls** displays because **cat** ignores the pipe
   b. **sort** displays the **ls** and **cat** displays the **wc**
   c. only the **wc** displays because **cat** ignores the pipe
   d. **cat** reads the pipe and the **wc** and displays both together
   e. nothing displays because **cat** ignores the pipe

5. If you are in **/etc** and **ls -l** shows a symbolic link **bar -> foo** then dereference the absolute path of **bar** with no symbolic links:
   a. **/bar/foo**   *b.* **/etc/bar/foo**   *c.* **/etc/foo/bar**
   d. **/etc/foo**   *e.* **/foo**

6. If file **a** contains 2 lines, and file **b** contains 3 lines, then how many lines are in file **c** after this command line: **ln a d ; ln d e ; ln b f >c**
   a. **3**   *b.* **5**   *c.* **2**   *d.* **4**   *e.* **0**

---

7. 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 32 is the inode number of this directory.
   b. The number 128 is the size of this directory.
   c. The number 128 is the count of links (names) this directory has.
   d. The number 128 is the inode number of this directory.
   e. The number 32 is the count of links (names) this directory has.

8. 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.* no such file (nonexistent)
   c. **bar**   *d.* nothing (empty file)
   e. **foo**

9. 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. **1**   *b.* **4**   *c.* **3**   *d.* **2**   *e.* **5**

10. 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. **/bin/bar/foo**   *b.* **/bin/bar**   *c.* **/bin/foo/bar**
    d. **/foo/bar**   *e.* **/bar**

11. Which command removes *only* this four-character name containing a special character: **?abc**
    a. **rm -r ?abc**   *b.* **rm "?abc"**   *c.* **rm ''?abc''**
    d. **rm /?abc**   *e.* **rm ""?abc""**

12. Which command shows the name of the current computer:
    a. **w**   *b.* **who**   *c.* **users**
    d. **hostname**   *e.* **comname**

13. Which option to **ls** displays the directory itself and not its contents?
    a. **-R**   *b.* **-d**   *c.* **-a**   *d.* **-i**   *e.* **-l**

14. What is the resulting link count of empty directory **dir** after these successful commands? **cd dir ; touch foo ; ln foo one ; ln foo two**
    a. **5**   *b.* **2**   *c.* **3**   *d.* **4**   *e.* **1**

15. Which of the following is true, given this long directory listing:
    drwxr-x--x 32 me me 128 Jan 1 1:00 dir
    a. The number 32 is the size 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 inode number of this directory.
    e. The number 128 is the count of links (names) this directory has.

16. 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. **bash: bat: command not found**
    b. **foo** followed by **hi**
    c. **foo**
    d. **hi**
    e. **hi** followed by **foo**

17. Which command line outputs inode/filename pairs for names in the current directory, sorted by inode number?
    a. **sort -n | ls -ai**   b. **ls -a | sort -i**
    c. **ls -ia > sort -n**   d. **ls -i -a | sort -n**
    e. **sort ls -ia**

18. What is true about this output from **ls -il foo bar**
    99 -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. **foo** and **bar** each have three names (six names total)
    d. **foo** and **bar** are names for the same file
    e. this output is not possible

19. 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**
    a. **one**
    b. **one** followed by **two**
    c. **bash: xxx: command not found**
    d. **two**
    e. **two** followed by **one**

20. If directory **dir** contains only these five two-character names: **a?**, **11**, **?1**, **1***, **.1**, then which command removes *only* the single two-character name **?1** from the directory?
    a. **rm dir/??**   b. **rm dir/\??**   c. **rm dir/*1**
    d. **rm dir/?1**   e. **rm dir/1***

21. In an empty directory, how many files are created by this command line:
    touch a "b c" ' ' d e
    a. **5**   b. **3**   c. **4**   d. **6**   e. **7**

22. 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]'**

23. What is the link count of directory **d** after these successful commands?
    mkdir d ; mkdir d/a d/b ; touch d/c d/e
    a. **1**   b. **4**   c. **3**   d. **5**   e. **2**

24. Rewrite as a simplified absolute path:
    /usr/./bin/../lib/../../etc/../usr/./lib/../bin/./bar
    a. **/bar**   b. **/usr/lib/bar**   c. **/usr/bin/bar**
    d. **/usr/bar**   e. **/etc/bar**

25. What displays on your screen given this command:
    date >date ; pwd >pwd ; head date | tail pwd
    a. **tail** reads the pipe and the **pwd** and displays both together
    b. nothing displays because **tail** ignores the pipe
    c. only the **pwd** displays because **tail** ignores the pipe
    d. **head** displays the **date** and **tail** displays the **pwd**
    e. only the **date** displays because **tail** ignores the pipe

26. 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**
    a. **bash: foo: command not found**
    b. **dad** followed by **mom**
    c. **dad**
    d. **mom**
    e. **mom** followed by **dad**

27. Which command line shows the current date?
    a. **echo date | bash**   b. **bash date**
    c. **bash >date ; cat date**   d. **bash <date**
    e. **date | bash**

28. What is the output on your screen after this two command sequence:
    PATH=/bin/ls:/bin/cat:/bin/sh ; cat nosuchfile
    a. **cat: nosuchfile: No such file or directory**
    b. **bash: /bin/ls: command not found**
    c. **bash: cat: command not found**
    d. **ls: /bin/cat: command not found**
    e. **bash: /bin/sh: No such file or directory**

29. How many arguments are passed to the command by the shell:
    <foo foo " a 'b c' d " e ' f " g " ' >foo
    a. **6**   b. **5**   c. **2**   d. **4**   e. **3**

30. If file **a** occupies one disk block, how many disk blocks are in use after this sequence of commands: `cp a b ; ln b c ; cp c d ; ln c e`
    a. **4**   b. **3**   c. **2**   d. **1**   e. **5**

31. How many arguments are passed to the command by the shell:
    `<wc wc " 1 ʼ2 3ʼ 4 " 5 6 ʼ 7 " 8 " ʼ >wc 9`
    a. **3**   b. **2**   c. **6**   d. **4**   e. **5**

32. Which command removes *only* this four-character name containing a special character: ***xyz**
    a. `rm ʼʼ*xyzʼʼ`   b. `rm ʼʼ*xyz`   c. `rm *xyz`
    d. `rm "*xyz"`   e. `rm *"xyz"`

33. Which of the following is true, given this long directory listing:
    `drwxr-x--x 512 me me 64 Jan 1 1:00 dir`
    a. The number 512 is the size of this directory.
    b. The number 64 is the inode number of this directory.
    c. The number 64 is the size of this directory.
    d. The number 512 is the inode number of this directory.
    e. The number 64 is the count of links (names) this directory has.

34. 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`
    a. **4**   b. **3**   c. **1**   d. **2**   e. **5**

35. 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/?`

36. Which command line displays the contents of file **foo** one page at a time?
    a. `less | foo`   b. `less <foo`
    c. `foo | less`   d. `cat foo >less`
    e. `foo >less`

37. In an empty directory, what is the output on your screen of this command line:
    `echo hi >foo ; cp foo bar | wc -l`
    a. **2**   b. **3**   c. **1**
    d. no output   e. **0**

38. What is the link count of directory **dir** after these successful commands?
    `mkdir dir ; touch foo ; cd dir ; ln ../foo bar`
    a. **1**   b. **2**   c. **3**   d. **4**   e. **5**

39. What is the output of this command line if run in an empty directory:
    `touch A a ; echo * >"*" ; ls`
    a. `* >*`   b. No output   c. `A a >A a`
    d. `* A a`   e. `A a >*`

40. Which command removes *only* this four-character name containing a special character: ***foo**
    a. `rm ?foo`   b. `rm .\/*foo`   c. `rm /*foo`
    d. `rm \*foo`   e. `rm \\*foo`

41. 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. **5**   b. **2**   c. **4**   d. **1**   e. **3**

42. Which of the following is true, given this long directory listing:
    `drwxr-x--x 32 me me 128 Jan 1 1:00 dir`
    a. The number 128 is the inode number of this directory.
    b. The number 32 is the count of links (names) this directory has.
    c. The number 128 is the count of links (names) this directory has.
    d. The number 32 is the size of this directory.
    e. The number 32 is the inode number of this directory.

43. How many arguments are passed to the command by the shell:
    `<pig pig -x " " -z -r" " >pig pig pig`
    a. **9**   b. **5**   c. **6**   d. **8**   e. **7**

44. What is true about this output from `ls -il foo bar`
    `23 -rwxr----- 3 me me 2 Jan 1 1:00 foo`
    `24 -rwxr----- 3 me me 2 Jan 1 1:00 bar`
    a. **foo** and **bar** each have two names (four names total)
    b. **foo** and **bar** are two of three names for this file
    c. **foo** and **bar** are names for different files
    d. this output is not possible
    e. **foo** and **bar** are names for the same file

45. What is the output on your screen after this two command sequence:
    `PATH=/bin/cat:/bin/sh:/bin/ls ; ls nosuchfile`
    a. `bash: ls: command not found`
    b. `ls: /bin/ls: command not found`
    c. `bash: /bin/sh: No such file or directory`
    d. `bash: /bin/ls: command not found`
    e. `ls: nosuchfile: No such file or directory`

46. What is the output on your screen after this command line:
    `echo hi >out | wc -l`
    a. **1**   b. **3**   c. no output
    d. **0**   e. **2**

47. 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`
    a. **0**   b. **4**   c. **3**   d. **2**   e. **1**

48. 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
    ```
    a. **4**   *b.* **1**   *c.* **5**   *d.* **2**   *e.* **3**

49. 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**
    a. **bash: xxx: command not found**
    b. **one** followed by **two**
    c. **one**
    d. **two**
    e. **two** followed by **one**

50. Which command removes *only* this two-character name containing a special character: **z?**
    a. **rm ?\?**   *b.* **rm "z?"**   *c.* **rm \z?**
    d. **rm \?**   *e.* **rm z?**

51. 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:
    a. **/bin/dir/bar**   *b.* **/foo/dir/bar**
    c. **/bin/foo/dir/bar**   *d.* **/bin/dir/bar/foo**
    e. **/dir/bar**

52. What is the link count of file **f** after these successful commands?
    ```
    rm f ; touch f ; cp f x
    ln f a ; ln x y ; ln a z ; ln z q
    ```
    a. **6**   *b.* **4**   *c.* **3**   *d.* **2**   *e.* **5**

53. Rewrite as a simplified absolute path:
    /home/me/../you/../../etc/../home/me/../you/../me/../foo
    a. **/home/you/foo**   *b.* **/home/foo**
    c. **/foo**   *d.* **/home/me/foo**
    e. **/etc/foo**

54. How many arguments are passed to the command by the shell:
    ```
    <f z " a 'b c' d " 1 2 ' g " h " ' >z
    ```
    a. **3**   *b.* **2**   *c.* **4**   *d.* **5**   *e.* **6**

55. How many arguments are passed to the command by the shell:
    ```
    <bat bat -b "-a -r" >bat bat bat
    ```
    a. **4**   *b.* **3**   *c.* **7**   *d.* **5**   *e.* **6**

56. 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/foo**   *b.* **/etc/bar/foo**   *c.* **/bar/foo**
    d. **/foo**   *e.* **/etc/foo/bar**

57. What is the link count of directory **d** after these successful commands?
    ```
    mkdir d ; mkdir d/a ; touch d/b
    ```
    a. **5**   *b.* **2**   *c.* **3**   *d.* **4**   *e.* **1**

58. Which of the following is true, given this long directory listing:
    ```
    drwxr-x--x 64 me me 512 Jan 1 1:00 dir
    ```
    a. The number 64 is the count of links (names) this directory has.
    b. The number 512 is the count of links (names) this directory has.
    c. The number 64 is the size of this directory.
    d. The number 512 is the inode number of this directory.
    e. The number 64 is the inode number of this directory.

59. 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 directory named **a/b/c**
    d. create a file named **a/b/c**
    e. create a hard link to directory **b** named **b2**

60. 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.* **4**   *c.* **5**   *d.* **3**   *e.* **2**

61. Given this **ls -il** long listing:
    ```
    298 drwxr-xr-x 296 me me 448 Jan 1 1:00 dir
    ```
    How many subdirectories lie immediately under **dir**?
    a. **294**
    b. **448**
    c. **296**
    d. **446**
    e. there is not enough information shown to answer the question

62. 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. **4**   *b.* **3**   *c.* **5**   *d.* **1**   *e.* **2**

63. What is true about this output from **ls -il foo bar**
    ```
    871 -r-------- 3 me me 2 Jan 1 1:00 foo
    872 -r-------- 3 me me 2 Jan 1 1:00 bar
    ```
    a. **foo** and **bar** each have two names (four names total)
    b. **foo** and **bar** each have three names (six names total)
    c. this output is not possible
    d. **foo** and **bar** are names for the same file
    e. **foo** and **bar** are two of three names for this file

64. 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. `10`   *b.* `6`   *c.* `18`   *d.* `2`   *e.* `21`

65. What is the link count of directory **foo** after these successful commands?
    `mkdir foo ; cd foo ; touch a b c`
    a. `2`   *b.* `3`   *c.* `4`   *d.* `5`   *e.* `1`

66. 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` followed by `1`   *b.* `2`   *c.* `1`
    d. `1` followed by `2`   *e.* no output

67. What is true about this output from `ls -il foo bar`
    `454 -rwxr-xr-x 3 me me 2 Jan 1 1:00 foo`
    `456 -rwxr-xr-x 3 me me 2 Jan 1 1:00 bar`
    a. **foo** and **bar** each have two names (four names total)
    b. **foo** and **bar** are two of three names for this file
    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

68. What is the output on your screen after this command line:
    `mkdir foo ; rmdir foo | wc -c`
    a. `1`   *b.* `4`   *c.* no output
    d. `0`   *e.* `3`

69. What is true about this output from `ls -il foo bar`
    `23 -r-x------ 2 me me 3 Jan 1 1:00 foo`
    `23 -r-x------ 2 me me 3 Jan 1 1:00 bar`
    a. **foo** and **bar** are names for the same file
    b. **foo** and **bar** each have three names (six names total)
    c. this output is not possible
    d. **foo** and **bar** are two of three names for the same file
    e. **foo** and **bar** are names for different files

70. If file **a** contains 2 lines, and file **b** contains 3 lines, then how many lines are in file **c** after this command line: `ln a d ; ln d c ; cat a b >c`
    a. `0`   *b.* `2`   *c.* `3`   *d.* `4`   *e.* `5`

71. How many arguments are passed to the command by the shell:
    `echo " 1 2 " three ' 4 ' five"6"`
    a. `4`   *b.* `5`   *c.* `3`   *d.* `9`   *e.* `1`

72. What is true about this output from `ls -il foo bar`
    `871 -r--------- 2 me me 3 Jan 1 1:00 foo`
    `871 -r--------- 2 me me 3 Jan 1 1:00 bar`
    a. **foo** and **bar** each have three names (six names total)
    b. **foo** and **bar** are names for the same file
    c. **foo** and **bar** are two of three names for this file
    d. this output is not possible
    e. **foo** and **bar** are names for different files

73. The option to `ls` that shows inode (index) numbers is:
    a. `-1`   *b.* `-i`   *c.* `-R`   *d.* `-l`   *e.* `-a`

74. How many arguments are passed to the command by the shell:
    `<cow cow "-x" -y '-z' >cow cow`
    a. `4`   *b.* `6`   *c.* `5`   *d.* `3`   *e.* `2`

75. What is the link count of directory **dir** after these successful commands?
    `mkdir dir ; cd dir ; touch foo ; mkdir a b c`
    a. `2`   *b.* `1`   *c.* `4`   *d.* `3`   *e.* `5`

76. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **a** (not in **c**) after this command line:
    `ln a d ; ln d c ; ln c e ; cat a b c d e >c`
    a. `3`   *b.* `11`   *c.* `9`   *d.* `2`   *e.* `5`

77. You enter this `cp a/b c/`
    and get `cp: a: No such file or directory`
    because:
    a. pathname **a** exists but is a file, not a directory
    b. directory **c** does not exist
    c. you forgot to specify the destination file name after **c/**
    d. the command **cp** is not in your search PATH
    e. directory **a** does not exist

78. What is the output on your screen after this command line:
    `echo hi >a ; cp a b | wc -c`
    a. `3`   *b.* `2`   *c.* `0`
    d. `1`   *e.* no output

79. 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`
    a. `6`   *b.* `2`   *c.* `9`   *d.* `5`   *e.* `3`

80. 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.* `7`   *c.* `5`   *d.* `3`   *e.* `2`

81. What is usually in the environment variable **$PATH**?
   a. the absolute path of your login shell
   b. the absolute path of your login home directory
   c. a colon-separated list of your **passwd** file fields
   d. a colon-separated list of directories containing command names
   e. the absolute path of the system **/path** directory

82. 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. **4**   b. **1**   c. **0**   d. **3**   e. **2**

83. What is true about this output from **ls -il foo bar**
   ```
   456 -rw-r--r-- 2 adm adm 3 Jan 1 1:00 foo
   456 -rwxr-xr-x 2 bin bin 3 Nov 9 2:55 bar
   ```
   a. **foo** and **bar** each have three names (six names total)
   b. **foo** and **bar** are names for the same file
   c. **foo** and **bar** are two of three names for this file
   d. this output is not possible
   e. **foo** and **bar** are names for different files

84. If file **foo** occupies two disk blocks, how many disk blocks are in use after this sequence of commands:
   ```
   cp foo bar ; ln bar one ; cp one two ; ln one ten
   ```
   a. **10**   b. **4**   c. **2**   d. **6**   e. **8**

85. 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. **1**   b. **3**   c. **2**   d. **4**   e. **5**

86. How many arguments are passed to the command by the shell:
   ```
   echo ' one two ' three ' four ' 5'6'
   ```
   a. **4**   b. **9**   c. **6**   d. **5**   e. **1**

87. Which one of these names is usually a shell environment variable?
   a. **foobar**   b. **Foobar**   c. **FOOBAR**
   d. **fooBar**   e. **FooBar**

88. 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**
   a. **two**
   b. **bash: /bin/xxx: command not found**
   c. **one**
   d. **one** followed by **two**
   e. **two** followed by **one**

89. 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/foo/bar**   b. **/etc/bar/foo**   c. **/etc/foo**
   d. **/bar/foo**   e. **/foo**

90. If I have a directory named **a/d**, which action would increase its *link count* by exactly one?
   a. create a directory named **a/d/e**
   b. create a file named **a/d2**
   c. create a directory named **a/d2**
   d. create a file named **a/d/e**
   e. create a hard link to directory **d** named **d2**

91. 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. **cat: bat: No such file or directory**
   b. **bash: cat: command not found**
   c. **bat**
   d. no output on screen
   e. **foo**

92. 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. **A** followed by **B**   b. nothing (empty file)
   c. **A**   d. **B**
   e. no such file (nonexistent)

93. 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
   ```
   a. **/bin/cat: foo: No such file or directory**
   b. **bash: /bin/cat: command not found**
   c. **bash: cat: command not found**
   d. no output on screen
   e. **bash: ls: command not found**

94. What is the link count of directory **d** after these successful commands?
   ```
   mkdir d ; cd d ; touch a ; mkdir b c d
   ```
   a. **2**   b. **5**   c. **6**   d. **3**   e. **4**

95. What is the link count of directory **dir** after these successful commands?
   ```
   mkdir dir ; mkdir dir/foo ; touch dir/bar
   ```
   a. **5**   b. **3**   c. **1**   d. **4**   e. **2**

96. What is the link count of file **f** after these successful commands?
   ```
   rm f ; touch f ; ln f bar
   cp bar x ; ln x y ; ln bar z ; ln z a
   ```
   a. **2**   b. **1**   c. **3**   d. **5**   e. **4**

97. What is true about this output from **ls -il foo bar**
    ```
    454 -rwxr-xr-x 3 me me 2 Jan 1 1:00 foo
    456 -rwxr-xr-x 3 me me 2 Jan 1 1:00 bar
    ```
    a. **foo** and **bar** are two of three names for this file
    b. **foo** and **bar** are names for different files
    c. **foo** and **bar** each have two names (four names total)
    d. **foo** and **bar** are names for the same file
    e. this output is not possible

98. What is in the local variable **$$** ?
    a. the first argument of the previous command line
    b. the process ID of the current shell
    c. the command name of the previous command line
    d. the cpu cost of the current session, in dollars
    e. **$$** is not a valid variable name

99. What is the link count of directory **d** after these successful commands?
    ```
    mkdir d ; cd d ; touch f ; ln f x ; ln f y
    ```
    a. **4**   b. **1**   c. **5**   d. **3**   e. **2**

100. Rewrite as a simplified absolute path:
    ```
    /../../var/./a/../../var/b/../../etc/./bar/../foo
    ```
    a. **/etc/bar/foo**   b. **/var/b/foo**   c. **/var/foo**
    d. **/var/a/foo**   e. **/etc/foo**

101. In an empty directory, what is the output on your screen after this command line:
    ```
    touch a ; ls | wc -l
    ```
    a. **1**   b. **0**   c. no output
    d. **3**   e. **2**

102. If your **PATH** variable contains **/bin:/usr/bin**, what is the output of this command line: **echo '$PATH'**
    a. **$PATH**
    b. **echo: $PATH: No such file or directory**
    c. **/bin:/usr/bin**
    d. **'$PATH'**
    e. **'/bin:/usr/bin'**

103. 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
    ```
    a. **foo** and **bar** are two of three names for this file
    b. this output is not possible
    c. **foo** and **bar** each have two names (four names total)
    d. **foo** and **bar** each have three names (six names total)
    e. **foo** and **bar** are names for the same file

104. 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** each have three names (six names total)
    b. this output is not possible
    c. **foo** and **bar** are names for different files
    d. **foo** and **bar** are names for the same file
    e. **foo** and **bar** are two of three names for the same file

105. 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 ten
    ```
    a. **1**   b. **4**   c. **3**   d. **2**   e. **5**

106. If I have a directory named **me/dir**, which action would increase its *link count* by exactly one?
    a. create a directory named **me/dir/.**
    b. create a directory named **me/dir2**
    c. create one file named **me/dir2**
    d. create one file named **me/dir/bar**
    e. create a directory named **me/dir/foo**

107. 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"**

108. The output of the **whoami** command is:
    a. the current directory
    b. a list of accounts in the password file
    c. your userid
    d. a list of users logged in to the system
    e. your HOME directory

109. 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. **5**   b. **1**   c. **4**   d. **2**   e. **3**

110. 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/a*"**   b. **rm /a/a?**   c. **rm /a/***
    d. **rm /a***   e. **rm /a/a***

111. How do you execute the program **foo** in the current directory?
    a. **foo/**   b. **./foo**   c. **$HOME/foo**
    d. **foo/.**   e. **/foo**

112. Which command line allows programs in the current directory to execute without preceding the names with `./`?
    a. `$PATH=/usr/bin:./bin`   *b.* `$PATH=.:$HOME:/usr/bin`
    c. `PATH=./$HOME:/usr/bin`   *d.* `PATH=/usr/bin:.:/bin`
    e. `PATH=/usr/bin/.:$HOME`

113. What is true about this output from `ls -il foo bar`
    `23 -rwxrwxrwx 2 me me 3 Jan 1 1:00 foo`
    `99 -rwxrwxrwx 2 me me 3 Jan 1 1:00 bar`
    a. `foo` and `bar` each have three names (six names total)
    b. this output is not possible
    c. `foo` and `bar` are names for the same file
    d. `foo` and `bar` are names for different files
    e. `foo` and `bar` are two of three names for the same file

114. What is the link count of an empty directory?
    a. `1`   *b.* `2`   *c.* `4`   *d.* `0`   *e.* `3`

115. How many arguments are passed to the command by the shell:
    `echo "cow "y " bat 'man x' " pig'a "hop' a b`
    a. `7`   *b.* `6`   *c.* `11`   *d.* `4`   *e.* `5`

116. Which command line would show the index (inode) number of a file?
    a. `ls -l file`   *b.* `ls -i file`   *c.* `cat -l file`
    d. `find -i file`   *e.* `cat -i file`

117. Which command removes *only* this four-character name containing a special character: `abc*`
    a. `rm abc\\*`   *b.* `rm abc/*`   *c.* `rm abc//*`
    d. `rm abc\*`   *e.* `rm abc*`

118. If you are in `/etc` and `ls -l` shows a symbolic link `bar -> ../you/foo` then dereference the absolute path of `bar` with no symbolic links:
    a. `/you/foo`   *b.* `/etc/you/foo`
    c. `/etc/bar/you/foo`   *d.* `/bar/you/foo`
    e. `/etc/you/foo/bar`

119. The option to `ls` that shows inode (index) numbers is:
    a. `-l`   *b.* `-I`   *c.* `-i`   *d.* `-x`   *e.* `-a`

120. 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. `foo` followed by `bar`
    b. `bar`
    c. `foo`
    d. `bar` followed by `foo`
    e. `bash: bat: command not found`

121. Which of the following is true, given this long directory listing:
    `drwxr-x--x 512 me me 64 Jan 1 1:00 dir`
    a. The number 512 is the count of links (names) this directory has.
    b. The number 64 is the inode number of this directory.
    c. The number 512 is the size of this directory.
    d. The number 512 is the inode number of this directory.
    e. The number 64 is the count of links (names) this directory has.

122. 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. `4`   *b.* `3`   *c.* `2`   *d.* `1`   *e.* `5`

123. What is in file `foo` after this command line:
    `echo hi >a ; ln a b ; echo me >b ; ln a foo ; rm a b`
    a. no such file (nonexistent)   *b.* `hi`
    c. nothing (empty file)   *d.* `hi` followed by `me`
    e. `me`

124. If I have a directory named `foo/bar`, which action would increase its *link count* by exactly one?
    a. create a file named `foo/cat`
    b. create a hard link to directory `bar` named `pig`
    c. create a directory named `foo/pig`
    d. create a directory named `foo/bar/9`
    e. create a file named `foo/bar/dog`

125. 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`
    a. `two` followed by `one`
    b. `one` followed by `two`
    c. `two`
    d. `bash: foo: command not found`
    e. `one`

126. 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. `bash: /bin/cat: command not found`
    b. `/bin/cat: pig: No such file or directory`
    c. `pig`
    d. no output on screen
    e. `bar`

127. What is the link count of file `a` after these successful commands?
    `ln a d ; cp a f ; ln d c ; ln f g ; ln c e`
    a. `1`   *b.* `2`   *c.* `3`   *d.* `5`   *e.* `4`

128. In an empty directory, what is in file **out** after this command line:
    ls nosuchfile | wc -l >out
    a. nothing (empty file)   *b.* **out**
    c. **1**   *d.* **nosuchfile**
    e. **0**

129. How many arguments are passed to the command by the shell:
    echo " one '2 three' 4 "five 6 ' 7 "8 ' >out
    a. **5**   *b.* **3**   *c.* **2**   *d.* **6**   *e.* **4**

130. What is usually in the environment variable **$PATH**?
    a. the absolute path of your login shell
    b. a colon-separated list of your **passwd** file fields
    c. the absolute path of your login home directory
    d. the absolute path of the system **/path** directory
    e. a colon-separated list of directories, each containing command files

131. 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**
    a. **mom**
    b. **hi**
    c. **bash: foo: command not found**
    d. **hi** followed by **mom**
    e. **mom** followed by **hi**

132. If file **a** contains 2 lines, and file **b** contains 3 lines, then how many lines are in file **c** after this command line: **ln a d ; ln b e ; cp d e >c**
    a. **0**   *b.* **3**   *c.* **5**   *d.* **2**   *e.* **4**

133. What is true about this output from **ls -il foo bar**
    72 -rwxrwxrwx 2 me me 3 Jan 1 1:00 foo
    72 -r--r--r-- 2 me me 3 Jan 1 1:00 bar
    a. **foo** and **bar** are two of three names for this file
    b. **foo** and **bar** are names for different files
    c. this output is not possible
    d. **foo** and **bar** are names for the same file
    e. **foo** and **bar** each have two names (four names total)

134. 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/bar/dir/foo**
    c. **/bin/dir/foo**   *d.* **/bar/../dir/foo**
    e. **/bin/dir/foo/bar**

135. What is the link count of file **f** after these successful commands?
    rm f ; touch f ; cp f x
    ln f a ; ln x y ; ln a z ; ln x b
    a. **3**   *b.* **2**   *c.* **4**   *d.* **5**   *e.* **6**

---

136. Which of the following is true, given this long directory listing:
    drwxr-x--x 64 me me 512 Jan 1 1:00 dir
    a. The number 512 is the inode number of this directory.
    b. The number 64 is the size of this directory.
    c. The number 512 is the size of this directory.
    d. The number 512 is the count of links (names) this directory has.
    e. The number 64 is the inode number of this directory.

137. What is usually in the environment variable **$SHELL**?
    a. the relative path of the **/home/shell** directory
    b. the absolute path of your login shell
    c. the relative path of the system **/shell** directory
    d. the absolute path of the system **/shell** directory
    e. the relative path of your login shell

138. What is in file **c** after this command line:
    echo A >a ; ln a b ; echo B >b ; ln a c ; rm a b
    a. **B**   *b.* **A** followed by **B**
    c. no such file (nonexistent)   *d.* **A**
    e. nothing (empty file)

139. How many arguments are passed to the command by the shell:
    echo " 1 '2 3' 4 "5 6 ' 7 "8 ' >out
    a. **5**   *b.* **2**   *c.* **3**   *d.* **4**   *e.* **6**

140. Given this **ls -il** long listing:
    302 drwxr-xr-x 202 me me 102 Jan 1 1:00 dir
    How many subdirectories lie immediately under **dir**?
    a. **100**   *b.* **200**   *c.* **102**   *d.* **202**   *e.* **300**

141. 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. **mom**
    b. **dad**
    c. **mom** followed by **dad**
    d. **bash: foo: command not found**
    e. **dad** followed by **mom**

142. 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**
    a. **two**
    b. **one**
    c. **two** followed by **one**
    d. **bash: foo: command not found**
    e. **one** followed by **two**

143. 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** followed by **xx**
    b. **foo**
    c. **xx**
    d. **bash: pig: command not found**
    e. **xx** followed by **foo**

144. 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 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

145. How many arguments are passed to the command by the shell:
    <cow cow "-x "-y '-z' >cow cow
    a. **7**   b. **5**   c. **6**   d. **4**   e. **3**

146. 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**

147. If the file **pig** 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 pig
    a. **/bin/ls: pig: No such file or directory**
    b. **pig**
    c. **foo**
    d. no output on screen
    e. **bash: /bin/ls: command not found**

148. 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"**

149. The correct syntax to assign to a shell variable is:
    a. **x=hello there**   b. **x = "hello there"**
    c. **x = hello there**   d. **x="hello there"**
    e. **"x=hello there"**

150. What is the output on your screen after this command line:
    echo one >x ; ln x y ; echo two >>y ; sort x
    a. **one**   b. **one** followed by **two**
    c. **two**   d. **two** followed by **one**
    e. no output

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

152. What is the link count of directory **d** after these successful commands?
    mkdir d ; mkdir d/a ; mkdir d/a/b ; mkdir d/a/c
    a. **5**   b. **3**   c. **1**   d. **4**   e. **2**

153. What command will recursively show disk usage in directories?
    a. **find**   b. **tree**   c. **ls**   d. **du**   e. **df**

154. 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. **two**   d. **one** followed by **two**
    e. **one**

155. In an empty directory, how many files are created by this command line:
    touch 1 "2 3" ' 4 ' 5
    a. **4**   b. **7**   c. **3**   d. **5**   e. **6**

156. Which command removes *only* this five-character name containing a special character: **yy?yy**
    a. **rm yy\?yy**   b. **rm yy\\?yy**   c. **rm yy/?yy**
    d. **rm yy?yy**   e. **rm yy//?yy**

157. 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
    a. **3**   b. **4**   c. **2**   d. **1**   e. **5**

158. 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. **6**   d. **5**   e. **4**

159. 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/a***   b. **rm /a/***   c. **rm /a/a?**
    d. **rm /a***   e. **rm /a/a***

160. Which command usually goes in your **`.bash_profile`** file?
   a. **`.bashrc source`**   *b.* **`.bash_profile source`**
   c. **`cat .bashrc`**   *d.* **`source ./.bash_profile`**
   e. **`source ./.bashrc`**

161. Which command line would show the inode (index) number of a file?
   a. **`cat -l file`**   *b.* **`find -i file`**   *c.* **`ls -i file`**
   d. **`cat -i file`**   *e.* **`ls -l file`**

162. What is true about this output from **`ls -il foo bar`**
   `816 drwxr-xr-x 2 me me 3 Feb 9 2:34 foo`
   `816 drwxr-xr-x 2 me me 3 Jan 1 1:00 bar`
   a. this output is not possible
   b. **foo** and **bar** are names for different files
   c. **foo** and **bar** are two of three names for this file
   d. **foo** and **bar** are names for the same file
   e. **foo** and **bar** each have three names (six names total)

163. Given this **`ls -il`** long listing:
   `298 drwxr-xr-x 448 me me 296 Jan 1 1:00 dir`
   How many subdirectories lie immediately under **dir**?
   a. **448**
   b. there is not enough information shown to answer the question
   c. **294**
   d. **446**
   e. **296**

164. 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 32 is the size 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 inode number of this directory.
   e. The number 32 is the count of links (names) this directory has.

165. 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`**

166. What is true about this output from **`ls -il foo bar`**
   `96 -rwxr-xr-x 2 me me 3 Jan 1 1:00 foo`
   `96 -rwxr-xr-x 3 me me 3 Jan 1 1:00 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. this output is not possible
   e. **foo** and **bar** are two of five names for this file

167. 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`**

168. Which command removes *only* this four-character name containing a special character: **`*dog`**
   a. **`rm \\*dog`**   *b.* **`rm .\/*dog`**   *c.* **`rm /*dog`**
   d. **`rm ?dog`**   *e.* **`rm ./\*dog`**

169. How many arguments are passed to the command by the shell:
   `<bar bar -b "-a" '-r' >bar bar bar`
   a. **6**   *b.* **5**   *c.* **4**   *d.* **3**   *e.* **7**

170. If I have a directory named **`/x/y/z`**, which action would increase its *link count* by exactly one?
   a. create one file named **`/x/y/z/x`**
   b. create a directory named **`/x/y/z2`**
   c. create one file named **`/x/y/z2`**
   d. create a directory named **`/x/y/z/.`**
   e. create a directory named **`/x/y/z/x`**

171. The output of the **`whoami`** command is:
   a. your HOME directory
   b. your userid
   c. a list of users logged in to the system
   d. the name of the current computer
   e. the current directory

172. Which command shows the name of the current computer:
   a. **`comname`**   *b.* **`hostname`**   *c.* **`history`**
   d. **`whoami`**   *e.* **`find`**

173. 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`
   a. no output on screen
   b. **bat**
   c. **`/bin/ls: bat: No such file or directory`**
   d. **`bash: /bin/ls: command not found`**
   e. **foo**

174. 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?"`**   *c.* **`rm /a\?`**
   d. **`rm /a/?\?`**   *e.* **`rm '/a/a?'`**

175. Given this **`ls -il`** long listing:
     `456 drwxr-xr-x 789 me me 123 Jan 1 1:00 dir`
     How many subdirectories lie immediately under **`dir`**?
     a. **`121`**
     b. **`123`**
     c. **`787`**
     d. there is not enough information shown to answer the question
     e. **`454`**

176. Which command line makes pathnames **`/usr/local/bin`** and **`/usr/bin`** lead to the same directory?
     a. **`ln . /usr/local`**   b. **`rmdir /usr/local`**
     c. **`mkdir /usr/local`**   d. **`ln -s . /usr/local`**
     e. **`touch /usr/local`**

177. What does *quoting* mean on a shell command line?
     a. typing a "control" character using the [CTRL] key
     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. using a leading tilde ("~") on a pathname to mean your HOME directory
     e. turning off the special meaning of shell meta-characters

178. What is usually in the environment variable **`$HOME`**?
     a. the relative path of your login home directory
     b. the absolute path of your login home directory
     c. the relative path of the ROOT directory
     d. the relative path of the system **`/home`** directory
     e. the absolute path of the system **`/home`** directory

179. In an empty directory, what is the output on your screen after this command line:
     `echo hi >a ; ls | wc -w`
     a. **`a`**   b. no output   c. **`2`**
     d. **`0`**   e. **`1`**

180. 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. **`1`**   c. **`4`**   d. **`3`**   e. **`2`**

181. What is the output on your screen after this command line:
     `echo 1 >x ; ln x y ; echo 2 >>y ; sort x`
     a. no output   b. **`2`**   c. **`2`** followed by **`1`**
     d. **`1`** followed by **`2`**   e. **`1`**

182. What is the link count of directory **`d`** after these successful commands?
     `mkdir  d  d/a  d/b  d/c  d/c/z ; touch  d/x  d/y`
     a. **`2`**   b. **`4`**   c. **`6`**   d. **`5`**   e. **`3`**

183. 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. **`5`**   c. **`3`**   d. **`2`**   e. **`6`**

184. Which command removes *only* this four-character name containing a special character: **`cat?`**
     a. **`rm "cat?"`**   b. **`rm ''cat?''`**   c. **`rm ""cat?""`**
     d. **`rm cat/?`**   e. **`rm \cat?`**

185. Which command removes *only* this four-character name containing a special character: **`*foo`**
     a. **`rm ?foo`**   b. **`rm \\*foo`**   c. **`rm .\/*foo`**
     d. **`rm ./\*foo`**   e. **`rm /*foo`**

186. How many arguments are passed to the command by the shell:
     `<bar bar -b"-a '-r' >bar" bar >out`
     a. **`4`**   b. **`5`**   c. **`3`**   d. **`2`**   e. **`6`**

187. How many arguments are passed to the command by the shell:
     `<foo foo " a 'b c' d " e ' f " g " ' >foo h`
     a. **`6`**   b. **`3`**   c. **`5`**   d. **`4`**   e. **`2`**

188. What is the link count of directory **`d`** after these successful commands?
     `mkdir d ; touch f ; cd d ; ln ../f x`
     a. **`4`**   b. **`3`**   c. **`1`**   d. **`2`**   e. **`5`**

189. 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 xxx`
     a. **`1`**   b. **`5`**   c. **`2`**   d. **`3`**   e. **`4`**

190. What is true about this output from **`ls -il foo bar`**
     `861 -rw-r--r-- 2 bin bin 3 Jan 1 1:00 foo`
     `861 -rw-r--r-- 2 adm adm 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 this 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)

191. A "dangling symlink" is a symlink:
     a. to the current directory   b. to a directory
     c. to a special device file   d. to a non-existent target
     e. to a parent directory

192. 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. `bash: prg: command not found`
    b. `hi` followed by `foo`
    c. `hi`
    d. `foo`
    e. `foo` followed by `hi`

193. 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. `a`   b. `2`   c. `0`
    d. no output   e. `1`

194. What is the link count of directory `a` after these successful commands?
    `mkdir a ; mkdir a/b ; mkdir a/c ; mkdir a/b/c`
    a. `2`   b. `5`   c. `4`   d. `3`   e. `1`

195. What is the link count of directory `dir` after these successful commands?
    `mkdir dir ; cd dir ; touch one ; mkdir two`
    a. `4`   b. `3`   c. `1`   d. `5`   e. `2`

196. What is true about this output from `ls -il foo bar`
    `816 -rw-r--r-- 2 me me 3 Jan 1 1:00 foo`
    `817 -rw-r--r-- 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 two of three names for this file
    d. `foo` and `bar` are names for the same file
    e. `foo` and `bar` are names for different files

197. If I have a directory named `/a/b/c/7`, which action would increase its *link count* by exactly one?
    a. create a directory named `/a/b/c/7e`
    b. create one file named `/a/b/c/7de`
    c. create a directory named `/a/b/c/d/e`
    d. create a directory named `/a/b/c/7/d2`
    e. create one file named `/a/b/c/7/d2`

198. Given this `ls -il` long listing:
    `454 drwxr-xr-x 123 me me 456 Jan 1 1:00 dir`
    How many subdirectories lie immediately under `dir`?
    a. `458`   b. `456`   c. `454`   d. `123`   e. `121`

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. `ten`   b. no output on screen
    c. `two`   d. `one` followed by `ten` and `two`
    e. `one`

200. If I have a directory named `/a/b/c/d`, which action would increase its *link count* by exactly one?
    a. create a directory named `/a/b/c/d2`
    b. create a directory named `/a/b/c/d/e`
    c. create one file named `/a/b/c/d/e`
    d. create one file named `/a/b/c/d2`
    e. create a directory named `/a/b/c/d`

201. If the file `foo` contained the word `mom`, what would be the output on your screen of this two command sequence:
    `PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/ls foo`
    a. `mom`
    b. `bash: /bin/ls: command not found`
    c. `foo`
    d. no output on screen
    e. `/bin/ls: foo: No such file or directory`

202. How many arguments are passed to the command by the shell:
    `echo 'And it's not hard, it's just logical.'`
    a. `7`   b. `5`   c. `3`   d. `6`   e. `4`

203. How many arguments are passed to the command by the shell:
    `echo " 1 2 "three ' 4 ' five"6"`
    a. `1`   b. `4`   c. `3`   d. `9`   e. `5`

204. Which of these statements is true?
    a. Only double quotes are strong enough to stop GLOB patterns from expanding.
    b. Only single quotes are strong enough to stop GLOB patterns from expanding.
    c. Only backslashes are strong enough to stop GLOB patterns from expanding.
    d. If `/x` is an empty directory, `sort /x/*` produces an error message.
    e. If `/y` is an empty directory, `echo /y/*` produces an error message.

205. What is the link count of directory `z` after these successful commands?
    `mkdir z ; cd z ; touch a b ; mkdir c d e`
    a. `6`   b. `7`   c. `5`   d. `3`   e. `4`

206. How many arguments are passed to the command by the shell:
    `echo 'It's "1 2" isn't it? I can't decide.`
    a. `6`   b. `2`   c. `5`   d. `4`   e. `3`

207. Which command removes *only* this four-character name containing a special character: **xyz?**
    a. **rm xyz\\?**   *b.* **rm xyz/?**   *c.* **rm -r xyz?**
    d. **rm xyz//?**   *e.* **rm xyz\?**

208. What is true about this output from **ls -il foo bar**
```
454 -rwxr-xr-x 2 me me 3 Jan 1 1:00 foo
454 -rwxr-xr-x 2 me me 3 Jan 1 1:00 bar
```
    a. **foo** and **bar** each have two names (four names total)
    b. **foo** and **bar** are two of three names for this file
    c. **foo** and **bar** are names for different files
    d. this output is not possible
    e. **foo** and **bar** are names for the same file

209. What is true about this output from **ls -il foo bar**
```
23 -r-x------ 2 bin bin 3 Jan 1 1:00 foo
23 -rwxrwxrwx 2 adm adm 3 Nov 9 9:59 bar
```
    a. **foo** and **bar** are names for different files
    b. **foo** and **bar** are names for the same file
    c. **foo** and **bar** are two of three names for the same file
    d. this output is not possible
    e. **foo** and **bar** each have three names (six names total)

210. 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/a***   *b.* **rm /a/a?**   *c.* **rm /a/a[*]**
    d. **rm /a/a\?**   *e.* **rm /a?**

211. 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. **787**   *b.* **123**   *c.* **456**   *d.* **789**   *e.* **454**

212. What is the output of this command line if run in an empty directory:
```
touch A a ; echo * ">*"
```
    a. **A a >A a**   *b.* *** >***   *c.* **A a**
    d. No output   *e.* **A a >***

213. If you are in **/etc** and **ls -l** shows a symbolic link **bar -> dir/foo** then dereference the absolute path of **bar** with no symbolic links:
    a. **/etc/dir/foo**   *b.* **/etc/dir/foo/bar**
    c. **/etc/bar/dir/foo**   *d.* **/dir/foo**
    e. **/bar/dir/foo**

214. In an empty directory, what is in file **count** after this command line:
```
ls ??? | wc -w >count
```
    a. nothing (empty file)   *b.* **1 1 2**
    c. **1 1 1**   *d.* **1**
    e. **0**

215. What is the link count of directory **d** after these successful commands?
```
mkdir d ; cd d ; touch f ; ln f a ; ln f b
```
    a. **3**   *b.* **1**   *c.* **2**   *d.* **4**   *e.* **5**

216. 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.* **4**   *d.* **2**   *e.* **3**

217. Which command line outputs inode/filename pairs for names in the current directory, sorted by inode number?
    a. **ls -i * > sort -n**   *b.* **sort -n | ls -ai**
    c. **ls -node * > sort -n**   *d.* **ls ./* | sort -node**
    e. **ls -ai | sort -n**

218. Which of the following **PATH** statements makes the most sense?
    a. **PATH=/bin:/etc/passwd:/usr/bin**
    b. **PATH=/bin/ls:/etc/passwd:/usr/bin**
    c. **PATH=/bin/bash:/usr/bin:/bin**
    d. **PATH=/bin:/usr/bin**
    e. **PATH=/bin:/usr/bin:/etc/passwd**

219. If I have a directory named **/1/2**, which action would increase its *link count* by exactly one?
    a. create a directory named **/1/2/3**
    b. create one file named **/1/2/3**
    c. create a directory named **/1/22**
    d. create one file named **/1/22**
    e. create a directory named **/1/2**

220. 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**
    a. **one** followed by **two**
    b. **one**
    c. **two** followed by **one**
    d. **two**
    e. **bash: /bin/foo: command not found**

221. What is in file **out** after this command line:
    echo me >a ; ln a b ; echo hi >b ; ln a out ; rm a b
  a. **me** followed by **hi**   *b.* **me**
  c. **hi**   *d.* no such file (nonexistent)
  e. nothing (empty file)

222. 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 '$cow'**
  a. the file names **1** through **8**
  b. the file names **1** through **8**, surrounded by quotes
  c. *****
  d. **$cow**
  e. **'$cow'**

223. 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**
  a. **foo** followed by **hi**
  b. **hi**
  c. **hi** followed by **foo**
  d. **foo**
  e. **bash: pig: command not found**

224. 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**
  b. **mom**
  c. **mom** followed by **dad**
  d. **bash: /bin/foo: command not found**
  e. **dad** followed by **mom**

225. How many arguments are passed to the command by the shell:
    echo 'It's a bird! It's a plane!'
  a. **4**   *b.* **2**   *c.* **5**   *d.* **3**   *e.* **1**

226. If you want a user-defined alias in all your **bash** shells, what do you do?
  a. put the alias into the **/etc/passwd** 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 **grub.conf** file for next log in
  e. put the alias into the **/etc/group** file for next log in

227. If your terminal type is **xterm**, what is the output of this command line?
    echo '$TERM'
  a. **'xterm'**   *b.* no output on screen
  c. **$TERM**   *d.* **xterm**
  e. **'$TERM'**

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

229. To change to the parent directory, do this:
  a. **pwd**   *b.* **pwd ..**   *c.* **cd .**
  d. **cd ..**   *e.* **cd**

230. Which command line allows programs in the current directory to execute without preceding the names with **./**?
  a. **$PATH=.:$HOME:/usr/bin**   *b.* **PATH=/bin:/usr/bin:.**
  c. **PATH=./$HOME:/usr/bin**   *d.* **PATH=/usr/bin/.:$HOME**
  e. **$PATH=/usr/bin:./bin**

231. What is the link count of directory **x** after these successful commands?
    mkdir x ; mkdir x/y ; mkdir x/z ; mkdir x/y/z
  a. **2**   *b.* **5**   *c.* **1**   *d.* **4**   *e.* **3**

232. Which command sorts *only* this five-character name containing a special character: **xx?xx**
  a. **sort "xx?xx"**   *b.* **sort ""xx?xx""**
  c. **sort ''xx?xx''**   *d.* **sort xx?xx**
  e. **sort xx/?xx**

233. 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.* **4**   *d.* **3**   *e.* **0**

234. 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.* **5**   *e.* **3**

235. 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. **5**   *b.* **3**   *c.* **4**   *d.* **2**   *e.* **1**

236. How many arguments are passed to the command by the shell:
```
<foo foo -x " " -z -r" " >foo 'foo foo'
```
a. **5**   *b.* **9**   *c.* **7**   *d.* **6**   *e.* **8**

237. 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
```
a. **foo** and **bar** each have three names (six names total)
b. **foo** and **bar** are names for different files
c. **foo** and **bar** are names for the same file
d. this output is not possible
e. **foo** and **bar** are two of three names for this file

238. What is true about this output from **ls -il foo bar**
```
454 -rwxr-xr-x 3 me me 2 Jan 1 1:00 foo
454 -rw-r--r-- 3 me me 2 Jan 1 1:00 bar
```
a. **foo** and **bar** are names for the same file
b. **foo** and **bar** are two of three names for this file
c. this output is not possible
d. **foo** and **bar** are names for different files
e. **foo** and **bar** each have three names (six names total)

239. If file **a** contains 2 lines, and file **b** contains 3 lines, then how many lines are in file
c after this command line:
```
ln a e ; ln b d ; ln d c ; cat e b >c
```
a. **4**   *b.* **2**   *c.* **0**   *d.* **5**   *e.* **3**

240. What is the link count of file **f** after these successful commands?
```
rm f ; touch f ; ln f b ; cp f g
cp b a ; ln a d ; ln b c ; cp c g
```
a. **4**   *b.* **3**   *c.* **5**   *d.* **2**   *e.* **1**

241. **Did you read all the words of the test instructions on page one?**
a. **Tak** *(Yes - Polish)*   *b.* **Igen** *(Yes - Hungarian)*
c. **Sim** *(Yes - Portuguese)*   *d.* **Taip** *(Yes - Lithuanian)*
e. **Jes** *(Yes - Esperanto)*

This page intentionally left blank.