

PRINT Name: \_\_\_\_\_

LAB Section:

--	--	--

## One-Answer Multiple Choice 44 Questions

Weight 15%

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

1. [15/88] What is the output on your screen of this command line:  

```
mkdir a ; touch b1 a/b1 a/b2 ; find a -name b?
```

  - a. a/b1
  - b. b1 a/b1 a/b2
  - c. b1 b2
  - d. no output
  - e. a/b1 a/b2
2. [21/90] In an empty directory, how many words are in file **foo** after this command line: 

```
echo b .d >c >.out ; cp c d ; ls >foo
```

  - a. 2
  - b. 0
  - c. 4
  - d. 1
  - e. 3
3. [30/89] Which command line outputs **/bin/date**?
  - a. **which date**
  - b. **touch /bin/date**
  - c. **cat /bin/date**
  - d. **cd /bin ; ls date**
  - e. **cd /bin ; echo date**
4. [36/88] In **/home/abcd0001** using **ls -l** shows a symbolic link **foo** -> **/bin/ls** then dereference the absolute path of **foo** with no symbolic links:
  - a. **/home/abcd0001/bin/ls/foo**
  - b. **/home/abcd0001/foo/bin/ls**
  - c. **/bin/ls**
  - d. **/foo/bin/ls**
  - e. **/home/abcd0001/bin/ls**

5. [38/89] Which command line below shows only lines **5-10** of file **foo**?
  - a. **head -n 10 foo | tail -n 6**
  - b. **head -n 6 foo | tail -n 10**
  - c. **tail -n 15 foo | head -n 5**
  - d. **tail -n 10 foo | head -n 6**
  - e. **head -n 10 foo | tail -n 5**
6. [40/90] 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 ; foo**
  - a. **two** followed by **one**
  - b. **one** followed by **two**
  - c. **bash: foo: command not found**
  - d. **one**
  - e. **two**
7. [42/90] If my current directory is **/bin**, which of these pathnames is equivalent to the file name **/bin/bash**?
  - a. **./bin/bash**
  - b. **bash**
  - c. **../bash**
  - d. **../bin/bash/.**
  - e. **bin/bash**
8. [44/89] Dereference the following symlink **bar** into its equivalent absolute path: **ln -s ../../a/../foo /tmp/a/b/bar**
  - a. **/tmp/a/foo**
  - b. **/tmp/b/bar**
  - c. **/tmp/foo**
  - d. **/tmp/a/b/bar**
  - e. **/tmp/b/foo**
9. [44/90] 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 a e ; cp a d ; rm a
```

  - a. 1
  - b. 2
  - c. 3
  - d. 5
  - e. 4
10. [45/88] In an empty directory, what is the output on your screen after this command line: **echo >.bar foo ; echo .\***
  - a. **. .. .bar**
  - b. **.\***
  - c. an error message from **echo** saying **.\*** does not exist
  - d. **foo**
  - e. **.bar**

11. [46/87] Given this long listing:  
`drwxr-xr-x 296 me me 448 Dec 4 9:12 dir`  
 How many subdirectories lie immediately under `dir`?
- 446
  - 448
  - there is not enough information shown to answer the question
  - 296
  - 294
12. [46/90] In an empty directory, what is in file `out` after this command line:  
`cat foo | wc -w >out`
- `foo`
  - `out`
  - 0
  - nothing (empty file)
  - 1
13. [47/90] What is true about this output from `ls -il foo bar`?
- ```
15 -rwxrwxrwx 2 bin bin 3 Jul 31 12:33 foo
15 -r--r--r-- 2 bin bin 3 Jul 31 12:33 bar
```
- `foo` and `bar` are names for the same file
  - this output is not possible
  - `foo` and `bar` each have three names (six names total)
  - `foo` and `bar` are names for different files
  - `foo` and `bar` are two of three names for the same file
14. [47/89] Which line allows the shell to find the `assignment07check` command?
- `PATH=$PATH:assignment07check`
  - `PATH=whereis assignment07check`
  - `PATH=$PATH:~idallen/cst8207/18w/assignment07`
  - `PATH=assignment07check:$PATH`
  - `PATH=which assignment07check`
15. [50/89] Give the minimum number of directories in this pathname:  
`/a/b/c`
- 4
  - 1
  - 3
  - 5
  - 2

16. [51/89] 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=/home:/bin/foo:/usr ; foo`
- `one`
  - `bash: foo: command not found`
  - `one` followed by `two`
  - `two` followed by `one`
  - `two`
17. [53/90] 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`
- 0
  - 2
  - 4
  - 3
  - 5
18. [54/89] Create a symbolic link under `/tmp` named `bar` that has target `foo`:
- `ln -s '/tmp/foo' /tmp/bar`
  - `ln -s 'foo' '/tmp/bar'`
  - `ln -s /tmp/bar '/tmp/foo'`
  - `ln -s /tmp/bar 'foo'`
  - `ln -s bar/foo /tmp`
19. [54/88] 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 ; cp a c ; rm a b`
- 4
  - 3
  - 1
  - 2
  - 5
20. [56/88] If I am in directory `/tmp` and `sd` is an empty sub-directory, what is true after this command line:  
`mkdir bar ; touch foo ; mv foo bar/sd`
- the directory `sd` is still empty
  - the directory `bar` now contains a file named `foo`
  - the directory `sd` now contains a file named `foo`
  - the command fails because `bar/sd` is not a directory
  - the directory `sd` now contains a directory named `bar`
21. [56/89] What is the link count of directory `x` after this set of successful commands? `mkdir x ; mkdir x/a x/a/b x/a/c x/a/d`
- 5
  - 3
  - 2
  - 4
  - 6

22. [57/90] File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this command line:  
`cat a a >c ; sort b >>a ; cat c b >c a`  
 a. 10      b. 12      c. 0      d. 8      e. 7
23. [58/89] File **a** contains 2 lines, and file **b** contains 3 lines, then how many lines are in file **c** after this command line:  
`sort a b >c ; cat a >>b ; sort c b >c a`  
 a. 12      b. 7      c. 5      d. 8      e. 0
24. [58/89] In `/usr/bin` using `ls -l` shows a symbolic link **foo** -> `../d/bar` then dereference the absolute path of **foo** with no symbolic links:  
 a. `/usr/bin/foo/../d/bar`      b. `/usr/d/bar`  
 c. `/usr/bin/d/bar`      d. `/usr/foo/../d/bar`  
 e. `/usr/bin/d/bar/foo`
25. [59/89] How many arguments are passed to the command by the shell:  
`echo " 1 '2 3' 4 "55 66 ' 7 "8 '999 >out`  
 a. 6      b. 4      c. 2      d. 5      e. 3
26. [59/89] If a shell GLOB pattern fails to match anything, the shell:  
 a. gives an error message and does not execute  
 b. returns the closest match to the pattern  
 c. gives a warning message but continues  
 d. removes the pattern and passes nothing  
 e. passes the pattern unchanged to the command
27. [59/88] What is in file **c** after this command line:  
`echo hi >a ; ln a b ; echo me >b ; ln a c ; rm a b`  
 a. **hi** followed by **me**      b. nothing (empty file)  
 c. **me**      d. **hi**  
 e. no such file (nonexistent)
28. [62/89] File **a** occupies one disk block. How many disk blocks are in use after this sequence of commands:  
`cp a b ; ln b c ; ln c d ; ln a e ; rm a b c`  
 a. 4      b. 2      c. 3      d. 1      e. 5

29. [62/89] Which pathname almost always leads to the same file named:  
`/bin/ls`  
 a. `/bin/./bin/./ls`      b. `./bin/ls`  
 c. `./bin/./ls/`      d. `./bin/./ls`  
 e. `/bin/./ls`
30. [64/89] If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:  
`111 -rw-r--r-- 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. 3      b. 0      c. 2      d. 1      e. 4
31. [65/89] File **a** occupies one disk block. How many disk blocks are in use after this sequence of commands:  
`ln a b ; ln b c ; cp c d ; ln c e ; rm a b c d`  
 a. 4      b. 3      c. 2      d. 1      e. 5
32. [65/89] If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:  
`111 -rw-r--r-- 3 me me 1 Jan 1 1:00 a`  
`111 -rw-r--r-- 3 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. 3      b. 4      c. 0      d. 1      e. 2
33. [66/89] How many files are touched or created?  
`touch " 1 '2 3' 4 " 5 6 ' 7 "8 '`  
 a. 3      b. 2      c. 4      d. 6      e. 5
34. [67/89] What is the link count of directory **x** after this set of successful commands?  
`mkdir x ; cd x ; touch a ; ln a b ; mkdir c d`  
 a. 7      b. 6      c. 5      d. 3      e. 4

35. [67/89] What is true about this output from `ls -il foo bar`

```
23 -rwxrwxrwx 2 bin bin 3 Jul 31 12:33 foo
23 -rwxrwxrwx 2 bin bin 3 Jul 31 12:33 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 the same file
- d. `foo` and `bar` are names for the same file
- e. `foo` and `bar` each have three names (six names total)

36. [68/89] What is true about this output from `ls -il foo bar`

```
111 -rwxrwxrwx 2 bin bin 3 Jul 31 12:33 foo
222 -rwxrwxrwx 2 bin bin 3 Jul 31 12:33 bar
```

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

37. [69/89] If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:

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

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

38. [73/87] 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-- 2 me me 1 Jan 1 1:00 b
333 -rw-r--r-- 2 me me 1 Jan 1 1:00 c
444 -rw-r--r-- 1 me me 1 Jan 1 1:00 d
```

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

39. [74/89] If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:

```
111 -rw-r--r-- 1 me me 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. 0            b. 3            c. 1            d. 4            e. 2

40. [75/89] If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:

```
111 -rw-r--r-- 1 me me 1 Jan 1 1:00 a
222 -rw-r--r-- 3 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. 0            b. 4            c. 3            d. 1            e. 2

41. [77/87] A "dangling symlink" is a symlink to:

- a. a parent directory
- b. a non-existent target
- c. a special device file
- d. a directory
- e. the current directory

42. [77/90] File `a` contains 2 lines. File `b` contains 3 lines. How many lines are output on your screen by this command line: `cat a | cat b`

- a. 3
- b. 2
- c. 5
- d. 2 followed by 3
- e. 3 followed by 2

43. [77/89] How do you execute the program `bar` in the current directory?

- a. `/bar`
- b. `bar/`
- c. `bar/.`
- d. `$HOME/bar`
- e. `./bar`

44. [79/89] Rewrite as a simplified absolute path (assume all directories exist):

```
/lib/./bin/./usr/./../etc/./lib/./usr/./bin/./bar
```

- a. `/bar`
- b. `/lib/bin/bar`
- c. `/lib/usr/bar`
- d. `/lib/bar`
- e. `/etc/bar`