

PRINT Name: _____

LAB Section:

--	--	--

One-Answer Multiple Choice 45 Questions**Weight 10%**

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

1. [19/91] What is the output of this in an empty directory:

```
pwd >.pwd ; ls >.ls ; echo .?*

```

 - a. an error message from **echo** saying **.?* does not exist**
 - b. **.pwd .ls**
 - c. **.. .pwd .ls**
 - d. **.?***
 - e. **. .. .pwd .ls**
2. [21/89] If file **a** contains 2 lines, and file **b** contains 3 lines, then how many lines are output on your screen by this: **head b | echo a**
 - a. **3** followed by **2**
 - b. **3**
 - c. **3** followed by **1**
 - d. **1**
 - e. **2**
3. [24/90] In an empty directory, how many words are in file **c** after this:

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

```

 - a. **0**
 - b. **3**
 - c. **1**
 - d. **4**
 - e. **2**
4. [28/89] What is the output of this in an empty directory:

```
touch x y z x1 xyx bxy yd .yx ye ; echo [xy]?

```

 - a. **x x1 y yd ye**
 - b. **xyx**
 - c. **[xy]?**
 - d. **x1 yd ye**
 - e. an error message from **echo** saying **[xy] ? does not exist**

5. [28/89] Which of these will make file **z** contain all of the content of file **x** followed by all of the content of file **y**?
 - a. **cp x >z ; cp y >>z**
 - b. **cp x z ; cat y >>z**
 - c. **cp x y >z**
 - d. **cat x >z ; cp y >>z**
 - e. **cp x y >>z**
6. [32/91] In an empty directory, how many words are in file **c** after this:

```
touch a >x ; echo y >x ; mv b a >b ; ls >c

```

 - a. **0**
 - b. **3**
 - c. **1**
 - d. **2**
 - e. **4**
7. [35/90] How many arguments and options are there to the command:

```
wc -wc ls cd

```

 - a. Three command names and one argument with two options.
 - b. Three arguments, one of which contains two options.
 - c. Three arguments with no options.
 - d. Two arguments and one option.
 - e. Four arguments, one of which is a command name.
8. [36/90] If **mt** is an empty sub-directory, what is true after this:

```
touch foo ; mkdir dir ; mv foo dir/mt

```

 - a. the command fails because **dir/mt** is not a directory
 - b. the directory **mt** now contains a directory named **dir**
 - c. the directory **mt** is still empty
 - d. the directory **mt** now contains a file named **foo**
 - e. the directory **dir** now contains a file named **foo**
9. [37/91] In an empty directory, what is in file **c** after this:

```
echo nosuchfile >c ; ls nosuchfile >c

```

 - a. **nosuchfile**
 - b. **nosuchfile c**
 - c. **ls: cannot access nosuchfile**
 - d. nothing (empty file)
 - e. **c**

10. [38/91] Which displays all the non-hidden names in the current directory that contain the word **ian** (and no other names)?
- `echo *ian*`
 - `echo ?ian?`
 - `echo [ian]`
 - `echo [i][a][n]`
 - `echo *[ian]*`
11. [39/90] If I am in directory **/tmp** and **mt** is an empty sub-directory, what is true after this:
- ```
touch mt/foo ; mkdir mt/me ; cp mt/foo mt/../me
```
- the command fails because the name **mt/./me** does not exist
  - there is a second copy of the file **foo** in file **/tmp/me**
  - the directory **mt** is now empty
  - there is a second copy of the file **foo** in directory **mt**
  - the directory **mt** now contains only a file named **me**
12. [39/91] What is the correct syntax to redirect both standard output and standard error into the same output file?
- `cmd >out 2>&1`
  - `cmd 1>out 2>out`
  - `cmd 2>&1 >out`
  - `cmd 2>1 >out`
  - `cmd 1>out 2>1`
13. [40/89] Give the minimum number of directories in this valid pathname:  
**/usr/local/lib**
- 4
  - 1
  - 3
  - 2
  - 5
14. [40/89] In an empty directory, what is the output on your screen after this:  
`date >.a ; touch .b ; ls *`
- no output
  - `.a .b`
  - `*`
  - an error message from **ls** saying **\*** does not exist
  - `. .. .a .b`

15. [41/89] If **mt** is an empty sub-directory, what is true after this:  
`touch foo ; mkdir bar ; mv mt/../foo mt/../bar`
- the directory **mt** now contains a file named **foo**
  - the directory **mt** is still empty
  - the directory **mt** now contains a file named **bar**
  - the command fails because **mt/bar** is not a directory
  - the directory **bar** now contains a file named **foo**
16. [41/87] If **mt** is an empty sub-directory, what is true after this:  
`touch mt/foo ; mkdir foo ; mv mt/foo mt/../foo/me`
- the **mkdir** fails because **foo** already exists
  - the directory **mt** now contains only a file named **me**
  - there is a second copy of the file **foo** in the file named **me**
  - the directory **mt** is now empty
  - the command fails because the name **mt/./foo/me** does not exist
17. [43/91] What is the output on your screen after these two command lines:  
`echo hi >a ; cp a c ; echo go >>c`  
`head a >c ; head a >a ; cat a c`
- hi** followed by **hi**
  - hi**
  - go**
  - hi** followed by **go**
  - no output
18. [43/90] Which **CTRL** key will send an **EOF** to a command reading your keyboard, and take you back to the command prompt?
- ^U**
  - ^D**
  - ^C**
  - ^E**
  - ^R**
19. [44/90] In an empty directory, what is the output on your screen of this:  
`echo xx >foo >bar ; cat foo`
- no output
  - `cat: foo: No such file or directory`
  - `xx >bar`
  - `xx >foo >bar`
  - `xx`

20. [45/90] What command will recursively find all pathnames named **lib** in **bar**?
- a. `find bar -name 'lib'`      b. `find lib -name 'bar'`  
 c. `grep 'lib' bar`      d. `grep -name 'lib' bar`  
 e. `grep bar -name 'lib'`
21. [46/88] Which pathname almost always leads to the same file named: **/dev/null**
- a. `/dev/../dev/./null`      b. `/dev/null/../null`  
 c. `../dev/null`      d. `./dev/null`  
 e. `/dev/null/.`
22. [48/90] If file **a** contains 2 lines, and file **b** contains 3 lines, then how many lines are output on your screen by this: `cp a b | tail`
- a. 2      b. 2 followed by 3  
 c. no output on screen      d. 3 followed by 2  
 e. 3
23. [49/90] If file **a** contains 2 lines, and file **b** contains 3 lines, then how many lines are in file **c** after this:
- ```
cat a a >c ; cat c >>a ; cat c b >c a
```
- a. 9 b. 11 c. 8 d. 7 e. 10
24. [50/91] If file **nine** contains 9 lines, each of which is the one-digit line number of the line in the file (1 through 9), what is the output on your screen of this: `cat nine nine | head -n 11 | tail -n 1`
- a. 9 b. 3 c. 1 d. 8 e. 2
25. [50/90] In a directory that contains only the file **ls**, what happens after this:
- ```
cp ls wc
```
- a. a copy of the file named **ls** is in the file named **wc**  
 b. an empty file named **wc** is created  
 c. the command fails because the name **wc** does not exist  
 d. the command fails because **wc** is not a directory  
 e. there is only the file named **wc** in the directory now

26. [51/89] If file **a** contains 2 lines, and file **b** contains 3 lines, then how many lines are output on your screen by this: `head a | head b`
- a. 3      b. no output on screen  
 c. 2      d. 2 followed by 3  
 e. 3 followed by 2
27. [52/89] How do I search for the text string **string** in the paginated output from the **man** command on my screen?
- a. `help string`      b. `grep string`      c. `find string`  
 d. `/string`      e. `\string`
28. [52/89] What is in file **out** after this: `echo 1 2 >out 3`
- a. nothing (empty file)      b. 1 2 out 3  
 c. 1 2 3      d. 1 2  
 e. 1 2 >out 3
29. [57/91] Which displays all the non-hidden names in the current directory that contain the case-insensitive word **ian** (and no other names)?
- a. `echo *[IiAaNn]*`  
 b. `echo *IiAaNn*`  
 c. `echo *[ian|IAN|Ian|iAn|iaN|IAN|IaN|IAN]*`  
 d. `echo *[Ii][Aa][Nn]*`  
 e. `echo *[ian,IAN,Ian,iAn,iaN,IAn,IaN,IAN]*`
30. [59/89] If file **nine** contains 9 lines, each of which is the one-digit line number of the line in the file (1 through 9), what is the output on your screen of this: `sort nine nine | tail -n 6 | head -n 1`
- a. 5      b. 4      c. 6      d. 3      e. 7
31. [60/90] How many lines are in file **x** after this:
- ```
date >x ; date >>x ; date >x ; date >>x
```
- a. 3 b. 2 c. 1 d. 4 e. 0
32. [60/90] The shell expands a leading tilde (~) in a pathname (e.g. `~/foo`) to be:
- a. the directory **/root** b. the current directory
 c. the parent directory d. your HOME directory
 e. the ROOT directory

33. [60/91] What command shows all the lines in file **file** that contain the text **mom**?
- `fgrep 'mom' >file`
 - `fgrep 'file' mom`
 - `cat file | find 'mom'`
 - `cat file > fgrep 'mom'`
 - `fgrep 'mom' file`
34. [61/91] If my current directory is **/usr/lib**, which of these pathnames is equivalent to the file name **/bin/bash**?
- `./bash`
 - `../bin/bash`
 - `./bin/bash`
 - `/usr/lib/./bash`
 - `../../bin/./bash`
35. [62/91] What is the output of this successful command sequence?
- ```
cd /var/tmp ; mkdir foo ; mkdir bar ; pwd
```
- `/tmp/foo/bar`
  - `/bar`
  - `/var/tmp`
  - `/foo/bar`
  - `/var/tmp/foo/bar`
36. [62/91] What is the output on your screen after this:
- ```
mkdir d ; touch d/.a d/.b ; echo d/*
```
- no output
 - `d/.a d/.b`
 - `d/.` `d/..` `d/.a` `d/.b`
 - `d/a` `d/b`
 - `d/*`
37. [64/91] Given the pathname **/usr/local/lib**, the *basename* of this pathname is:
- `local`
 - `/`
 - `/usr/local`
 - `usr`
 - `lib`
38. [65/90] If my current directory is **/tmp**, which command copies the password file into the existing directory **/tmp/dir** under the name **bar**?
- `cp ../etc/passwd ./bar`
 - `cp ../dir/../../etc/passwd ../bar`
 - `cp dir/../../etc/passwd dir/../../tmp/dir/bar`
 - `cp ../../etc./passwd /dir/tmp/bar`
 - `cp ./etc/passwd dir/bar`

39. [65/91] Which deletes a directory and everything inside it?
- `rd -a dir`
 - `rmdir -p dir`
 - `del -p dir`
 - `rm -r dir`
 - `rmdir -r dir`
40. [67/90] In a manual page **SYNOPSIS** section, ellipsis (three dots) (`...`) mean:
- something that is optional
 - something that is repeated
 - no special meaning
 - the parent directory
 - a hidden directory
41. [67/90] The option to **ls** that shows hidden names is:
- `-h`
 - `-i`
 - `-a`
 - `-l`
 - `-1`
42. [70/91] To "throw away" (hide) standard error output of a command, use:
- `cmd 1>&2`
 - `cmd 1>/dev/sda1`
 - `cmd 2>/dev/null`
 - `cmd 2>&1`
 - `cmd 2>/dev/sda1`
43. [71/89] **Did you read all the words of the test instructions on page one?**
- Tak** (Yes - Polish)
 - Jes** (Yes - Esperanto)
 - Taip** (Yes - Lithuanian)
 - Sim** (Yes - Portuguese)
 - Igen** (Yes - Hungarian)
44. [75/91] Which **CTRL** key will send an **Interrupt** to make a command end and take you back to the command prompt?
- `^C`
 - `^D`
 - `^U`
 - `^E`
 - `^R`
45. [78/90] The command that creates a directory and all parent directories is:
- `mkdir -r a/b/c`
 - `mkdir -p a/b/c`
 - `pwd -r a/b/c`
 - `cp -p a/b/c`
 - `touch -p a/b/c`