PRINT Name: _____

One-Answer Multiple Choice **182 Questions** **Weight 40%**

☞ 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 question about reading the test instructions is: **123**

---

191. Answer **191** is **B**
192. Answer **192** is **C**
193. Answer **193** is **E**
194. Answer **194** is **A**
195. Answer **195** is **A**
196. Answer **196** is **D**

Your **Test Version** is:

B C E A A D

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

---

1. Given my directory **dir** and my file **dir/bar** which permissions allow me to delete the file from the directory, but not append data to the file?
   a. Permissions **500** on directory **dir** and **400** on file **dir/bar**.
   b. Permissions **300** on directory **dir** and **300** on file **dir/bar**.
   c. Permissions **100** on directory **dir** and **200** on file **dir/bar**.
   d. Permissions **300** on directory **dir** and **500** on file **dir/bar**.
   e. Permissions **100** on directory **dir** and **100** on file **dir/bar**.
2. Which of the following would result in a "true" exit status?
   a. `[ '00' != "00" ]` *b.* `[ 00 = 0 ]`
   c. `[ '00' -ne "0" ]` *d.* `[ '00' = "0" ]`
   e. `[ '00' -eq "0" ]`
3. Which command usually goes in your **.bash_profile** file?
   a. `./.bash_profile source` *b.* `./.bashrc source`
   c. `cat ./.bashrc` *d.* `source ./.bashrc`
   e. `source ./.bash_profile`
4. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
   ```
   dr-xrwx-wx 2 pat pgg 60 Jan 1 1:00 foo
   -r-xrwxr-x 1 bob bg2  0 Jan 1 1:00 foo/bar
   ```
   a. **bob** can create a new file in the directory
   b. **pat** can rename the file
   c. **bob** can access and write on the file
   d. **pat** can access and write on the file
   e. **bob** can list names in the directory

5. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
   ```
   d-w---xr-- 2 pat ted 60 Jan 1 1:00 foo
   -rwxrwxrwx 1 pat bg2  0 Jan 1 1:00 foo/bar
   ```
   a. **bob** can rename the file
   b. **bob** can access and write on the file
   c. **pat** can access and write on the file
   d. **bob** can create a new file in the directory
   e. **bob** can list names in the directory
6. Given the following, can user **bird** in group **sesame** append to **foobar**?
   ```
   drwx--xrwx 2 root sesame 4096 Oct 7 14:00 .
   -rw------- 1 bird sesame 1024 Oct 4 14:05 foobar
   ```
   a. Yes, because **bird** has write permissions on **foobar**
   b. No, because **sesame** has no write permissions on **foobar**
   c. No, because the directory is not accessible to **bird**
   d. Yes, because **bird** owns **foobar**
   e. No, because execute permissions are not set for **bird** on **foobar**
7. The shadow password file is used:
   a. to reduce the size of the main password file for faster access
   b. to hide encrypted passwords from viewing by ordinary users
   c. to store secondary passwords for times when you forget your main one
   d. to keep a back-up of the main password file in case of corruption
   e. to allow passwords to exist on partitions other than the ROOT
8. Inside a shell script, which correctly expands to be the first script argument without processing any special characters in the argument?
   a. `'$1'` *b.* `"\$1"` *c.* `\$1`
   d. `"$1"` *e.* `$1`
9. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
   ```
   dr-xrwx--x 2 pat pgg 60 Jan 1 1:00 foo
   --w----r-x 1 bob bg2  0 Jan 1 1:00 foo/bar
   ```
   a. **pat** can rename the file
   b. **bob** can list names in the directory
   c. **pat** can access and write on the file
   d. **bob** can access and write on the file
   e. **bob** can create a new file in the directory
10. What is the output on your screen of the following sequence of commands:
    ```
    x=pig ; [ -z $x ] ; echo $?
    ```
    a. **1**
    b. **0**
    c. no output
    d. the number 0 or 1 followed by another 0 or 1 on a new line
    e. `test: $x: integer expression expected`

11. Which test checks to see if the path is not an empty file (zero bytes)?
    a. `test -n path` *b.* `test -x path` *c.* `test -z path`
    d. `test -s path` *e.* `test -e path`

12. If you have a file `crontab.day` of commands in `crontab` format, you could submit that file to be your live `crontab` file by running which of the following commands?
    a. `crontab -l crontab.day`
    b. `crontab -e crontab.day`
    c. `crontab < crontab.day`
    d. `echo crontab.day | crond`
    e. `crontab > crontab.day`

13. Given the following shell script statement,
    `if [ "a" = "b" ] ; then echo SAME ; fi`
    which of the following statements is true?
    a. "`[`" is passed four arguments
    b. an "invalid number" error would result
    c. "`SAME`" would be printed
    d. "`[`" is part of all "`if`" statements
    e. "`fi`" would cause a "command not found" error

14. User `bob` is in groups `bg1` and `bg2`. User `pat` is in group `pgg`.
```
d--x-----x 2 pat pgg 60 Jan 1 1:00 foo
-r-xrwx-w- 1 bob bg1  0 Jan 1 1:00 foo/bar
```
    a. `bob` can list names in the directory
    b. `pat` can rename the file
    c. `bob` can create a new file in the directory
    d. `bob` can access and write on the file
    e. `pat` can access and write on the file

15. What is the output (if any) of this program fragment? (There are blanks between all the digits in the word list section of the `for` loop.)
```
s=0
for i in 1 2 3 4
do
    s=$((s+i))
done
echo "$s"
```
    a. `10` *b.* `1` *c.* `4321`
    d. `1234` *e.* `1 2 3 4`

16. User `bob` is in groups `bg1` and `bg2`. User `pat` is in group `pgg`.
```
drw-rw-rwx 2 pat bg1 60 Jan 1 1:00 foo
-rwxrwxrwx 1 pat ted  0 Jan 1 1:00 foo/bar
```
    a. `pat` can rename the file
    b. `pat` can create a new file in the directory
    c. `bob` can access and write on the file
    d. `bob` can rename the file
    e. `bob` can list names in the directory

17. Given my directory `dir` and my file `dir/bar` which permissions allow me to delete the file from the directory, but not append data to the file?
    a. Permissions `100` on directory `dir` and `300` on file `dir/bar`.
    b. Permissions `500` on directory `dir` and `500` on file `dir/bar`.
    c. Permissions `300` on directory `dir` and `400` on file `dir/bar`.
    d. Permissions `100` on directory `dir` and `500` on file `dir/bar`.
    e. Permissions `300` on directory `dir` and `200` on file `dir/bar`.

18. Which of the following signals is strongest (cannot be handled or ignored)?
    a. `SIGSUSP` *b.* `SIGHUP` *c.* `SIGINT`
    d. `SIGTERM` *e.* `SIGKILL`

19. What value to `chmod` would change the permissions on a file to `r-----rw-`?
    a. `654` *b.* `406` *c.* `322` *d.* `122` *e.* `102`

20. The **minimum** permissions you need to copy a file `foo` from directory `a` to directory `b` are:
    a. `rwx` on `a`, `wx` on `b`, none on `foo`
    b. `rx` on `a`, `wx` on `b`, `w` on `foo`
    c. `x` on `a`, `wx` on `b`, `r` on `foo`
    d. `wx` on `a`, `wx` on `b`, none on `foo`
    e. `wx` on `a`, `wx` on `b`, `rw` on `foo`

21. Given this successful command line (note the dot argument):
    `cd /tmp ; mkdir dir ; cd dir ; chmod u-x .`
    Which next command will execute without any "permission denied" errors?
    a. `ls /tmp/dir` *b.* `ls /tmp/dir/.`
    c. `ls .` *d.* `ls /tmp/dir/..`
    e. `ls ..`

22. Which command line makes a directory `dir` into which anyone can put a file, but in which nobody can see the names of the files that are there?
    a. `mkdir dir ; cd dir ; chmod ugo-rw .`
    b. `mkdir dir ; chmod 333 .`
    c. `mkdir dir ; cd dir ; chmod ugo=w .`
    d. `mkdir dir ; chmod 333 dir`
    e. `mkdir dir ; chmod 222 dir`

23. When a user named **bob** runs a command in an executable file owned by **foo**, in a directory owned by **root**, the file executes with the permissions of:
    a. **bob** *b.* **root and bob** *c.* **root**
    d. **root and foo** *e.* **foo**

24. Which of these statements is true?
    a. If you give me write permission on a file owned by you, I can then use **chmod** to change its permissions.
    b. To make a hard link to file "**foo**" named "**bar**", file "**foo**" must exist.
    c. You only need "**r--**" permission on directory "**foo**" for "**ls -l foo**" to work.
    d. The "**ln**" command takes two arguments, so the maximum number of hard links a file can have is two.
    e. You can make a hard link to a directory.

25. What value **umask** gives a new file permissions **r--r-----**?
    a. **326** *b.* **110** *c.* **447** *d.* **440** *e.* **220**

26. Can three different files have the same inode number on three different file systems?
    a. no: inode numbers are unique across all file systems
    b. no: you can't have inode numbers on three file systems
    c. yes: inode numbers are only unique inside a file system
    d. yes: if the files are all names for the same inode
    e. no: inode numbers only apply to directories, not files

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

28. Dereference the following symlink **bar** into its equivalent absolute path:
```
ln -s ../b/../../a/../foo /tmp/a/b/bar
```
    a. **/tmp/b/foo** *b.* **/tmp/b/bar** *c.* **/tmp/a/b/bar**
    d. **/tmp/foo** *e.* **/tmp/a/foo**

29. Given the following, can user **bird** in group **sesame** append to **foobar**?
```
drwxrw-rwx 2 root sesame 4096 Oct 7 14:00 .
-rw-rw-r-- 1 bird sesame 1024 Oct 4 14:05 foobar
```
    a. No, because execute permissions are not set for **bird** on **foobar**
    b. No, because the directory is not accessible to **bird**
    c. Yes, because **bird** owns **foobar**
    d. Yes, because **bird** has write permissions on **foobar**
    e. Yes, because **sesame** has write permissions on **foobar**

30. If I mount one file system on directory **/a** and another file system on directory **/b,** how can I link the existing file **/a/foo** to the new pathname **/b/new**?
    a. **ln /a/foo /b/new** *b.* **ln -s /a/foo /b/new**
    c. **ln /a/new /b/foo** *d.* **ln /b/new /a/foo**
    e. **ln -s /b/new /a/foo**

31. What is the output of this command line in an empty directory:
```
touch .a .b .c ; echo [.]*
```
    a. **[.]***
    b. no output
    c. **. .. .a .b .c**
    d. **.a .b .c**
    e. an error message from **echo** saying **[.]*** does not exist

32. In an empty directory, what is output on your screen by:
```
mkdir -p a/b/c 1/2/3 ; mv a/b/c 1/2 ; find . -name c
```
    a. **./1/a/b/c** *b.* **./1/2/a/b/c**
    c. **./1/2/b/c** *d.* **./1/2/c**
    e. **./1/2/3/a/b/c**

33. What permissions are given to **newfile** after this command line:
```
umask 326 ; touch newfile
```
    a. **-wx-w-rw-** *b.* **r--r-----** *c.* **r--r-x--x**
    d. **-wxr-----** *e.* **-wx-w-r-x**

34. If **a=123** and **b=456** then what is the output of the following sequence of commands: **if [$a = $b]; then echo $a ; fi**
    a. no output
    b. **test: $a: string expression expected**
    c. **test: a=123: integer expression expected**
    d. **123**
    e. **bash: 123: command not found**

35. Which expands to all the script arguments?
    a. **"$!"** *b.* **"$#"** *c.* **"$*"** *d.* **"$?"** *e.* **"$0"**

36. If the line, **exit 2**
    is executed in a shell script, what is the result?
    a. termination with an exit status of 0
    b. termination after sleeping for 2 seconds
    c. termination with an exit status of 2
    d. an invalid argument error message
    e. the script breaks out of up to 2 levels of loops

37. The **cron** system can run commands at most every
    a. hour *b.* second *c.* minute
    d. millisecond *e.* day

38. If a script named **bar** contains a loop that starts: **for i do**
and the script is executed using this command line:
./bar a ' b d ' e f " g h " a
how many times will the loop iterate?
a. 6 iterations *b.* 1 iteration *c.* 7 iterations
d. 9 iterations *e.* 8 iterations

39. If **bar** is an executable script containing the line **animal=dog** then what is the **bash** output of this sequence of three commands:
animal=pig ; ./bar ; echo "the '$animal' ate"
a. **the '$animal' ate** *b.* **the 'pig' ate**
c. **the 'animal' ate** *d.* **the $animal ate**
e. **the 'dog' ate**

40. In a directory containing one file named **dog**, what is the output on your screen after this command line: **1>/dev/null ls ***
a. *****
b. no output
c. **ls: *: No such file or directory**
d. **bash: 1>/dev/null: command not found**
e. **dog**

41. What does the **-v** option to the **fgrep** command do?
a. selects lines that do not contain a match for the supplied pattern
b. prints the version number of the **fgrep** command
c. turns off the translation of unprintable characters
d. selects lines that do not contain unprintable characters
e. turns on the translation of unprintable characters

42. A **crontab** entry of **0 6 * * * /sbin/somescript**
would run **somescript** when and how often?
a. at 6:00am every business day
b. at 12:06am every business day and Saturday
c. at 6:00am every day
d. at 12:06am every business day
e. at 12:06am every day

43. Process signals in increasing order of strength:
a. **TERM HUP KILL** *b.* **KILL HUP TERM**
c. **HUP KILL TERM** *d.* **HUP TERM KILL**
e. **TERM KILL HUP**

44. What value **umask** gives a new file permissions **r--r-----**?
a. **440** *b.* **220** *c.* **110** *d.* **237** *e.* **446**

45. The **minimum** permissions you need to append to a file **foo** in directory **a** are:
a. **rwx** on **a**, none on **foo** *b.* **x** on **a**, **w** on **foo**
c. **wx** on **a**, **w** on **foo** *d.* **wx** on **a**, none on **foo**
e. **rwx** on **a**, **rw** on **foo**

46. What is the output on your screen of the following command sequence:
i=04; test $i = 4 ; echo $?
a. **1**
b. no output
c. **test: $i: integer expression expected**
d. **0**
e. the number 0 or 1 followed by another 0 or 1 on a new line

47. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
d--x----w- 2 pat ted 60 Jan 1 1:00 foo
--w-r-xrwx 1 pat bg2 0 Jan 1 1:00 foo/bar
a. **pat** can access and write on the file
b. **bob** can list names in the directory
c. **bob** can access and write on the file
d. **pat** can rename the file
e. **bob** can create a new file in the directory

48. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
dr---wx--x 2 bob ted 60 Jan 1 1:00 foo
-r-xrwxrwx 1 pat bg1 0 Jan 1 1:00 foo/bar
a. **bob** can access and write on the file
b. **bob** can create a new file in the directory
c. **pat** can rename the file
d. **pat** can access and write on the file
e. **bob** can list names in the directory

49. Which command sequence correctly searches for the **string** and then prints **OK** if it is found inside the password file?
a. **if [fgrep string /etc/passwd] ; then echo OK ; fi**
b. **if test string /etc/passwd ; then echo OK ; fi**
c. **if fgrep string /etc/passwd ; then echo OK ; fi**
d. **if [test string /etc/passwd] ; then echo OK ; fi**
e. **if test string = /etc/passwd ; then echo OK ; fi**

50. To show all your one-time scheduled commands, type:
a. **atq** *b.* **cat crontab**
c. **/var/log/crontab** *d.* **crontab -l**
e. **/etc/crontab**

51. What is the output on your screen of the following sequence of commands:
i=00 ; [$i -eq 0] ; echo $?
a. **0**
b. the number 0 or 1 followed by another 0 or 1 on a new line
c. no output
d. **1**
e. **test: $i: integer expression expected**

52. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-w-rwx-wx 2 bob ted 60 Jan 1 1:00 foo
-r-xrwxrwx 1 pat bg2  0 Jan 1 1:00 foo/bar
```
a. **pat** can rename the file
b. **bob** can list names in the directory
c. **bob** can create a new file in the directory
d. **pat** can access and write on the file
e. **bob** can access and write on the file

53. Which command removes adjacent duplicate lines from a file?
a. **dupl** *b.* **uniq** *c.* **unique**
d. **duplicate** *e.* **dup**

54. Which **crontab** line executes at **13:54** every day?
a. **13 * * * 54 command** *b.* *** * * 13 54 command**
c. **13 54 * * * command** *d.* *** * * 54 13 command**
e. **54 13 * * * command**

55. What minimal permissions must you have on a directory to be able to execute successfully the command **ls .** from *inside* the directory?
a. **r--** *b.* **r-x** *c.* **-wx** *d.* **--x** *e.* **rw-**

56. If **a=123** and **b=456** then what is the output of the following sequence of commands: **if [$a = $b] ; then echo $a ; fi**
a. **test: $a: string expression expected**
b. **bash: [123: command not found**
c. **123**
d. no output
e. **test: a=123: integer expression expected**

57. To list your personal crontab, type:
a. **atq** *b.* **cat crontab**
c. **/var/log/crontab** *d.* **/etc/crontab**
e. **crontab -l**

58. Which of the following, as first line of a shell script, would mean that when the script is run as a command, **/bin/sh** will be run with the **-u** option to process the script.
a. **!#/bin/sh -u** *b.* **#!/bin/sh -u** *c.* **!/bin/sh -u**
d. **!!/bin/sh -u** *e.* **#/bin/sh -u**

59. What command line shows only your own processes, not all processes?
a. **ps lxww** *b.* **psmine** *c.* **crontab**
d. **showall** *e.* **dmesg**

60. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr-x-wx--x 2 bob ted 60 Jan 1 1:00 foo
-r-x-w-r-x 1 bob bg1  0 Jan 1 1:00 foo/bar
```
a. **pat** can rename the file
b. **bob** can access and write on the file
c. **bob** can create a new file in the directory
d. **pat** can access and write on the file
e. **bob** can list names in the directory

61. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr-xr-xrwx 2 pat bg1 60 Jan 1 1:00 foo
-rwxrwxr-x 1 pat ted  0 Jan 1 1:00 foo/bar
```
a. **bob** can access and write on the file
b. **pat** can create a new file in the directory
c. **bob** can rename the file
d. **pat** can rename the file
e. **bob** can list names in the directory

62. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-wx-w-rwx 2 pat bg2 60 Jan 1 1:00 foo
-rwxrwxrwx 1 pat ted  0 Jan 1 1:00 foo/bar
```
a. **bob** can access and write on the file
b. **pat** can rename the file
c. **bob** can rename the file
d. **bob** can create a new file in the directory
e. **bob** can list names in the directory

63. If variable **a** might contain nothing (a null value - defined but empty), which command sequence correctly tests for this and prints the date?
a. **if ['''' = $a] ; then date ; fi**
b. **if ["$a" = *] ; then date ; fi**
c. **if test "" -eq $a ; then date ; fi**
d. **if test "" = "$a" ; then date ; fi**
e. **if [$a = /dev/null] ; then date ; fi**

64. Which of these commands makes a file owned by me, also readable by me?
a. **chmod r+u myfile** *b.* **umask 400 myfile**
c. **chmod u+r ./myfile** *d.* **chmod r=u ./myfile**
e. **umask 300 ./myfile**

65. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d--xr----x 2 bob ted 60 Jan 1 1:00 foo
-r-x-w-rwx 1 pat bg2  0 Jan 1 1:00 foo/bar
```
a. **bob** can access and write on the file
b. **bob** can list names in the directory
c. **bob** can create a new file in the directory
d. **pat** can access and write on the file
e. **pat** can rename the file

66. Given my directory **dir** and my file **dir/bar** which permissions allow me to access and append data to the file but not delete the file?
    a. Permissions **200** on directory **dir** and **200** on file **dir/bar**.
    b. Permissions **400** on directory **dir** and **400** on file **dir/bar**.
    c. Permissions **600** on directory **dir** and **700** on file **dir/bar**.
    d. Permissions **100** on directory **dir** and **600** on file **dir/bar**.
    e. Permissions **500** on directory **dir** and **100** on file **dir/bar**.

67. To change your own account password, use this exact command line:
    a. **$ passwd**
    b. **$ passwd .**
    c. **$ passwd ***
    d. **$ passwd cst8207**
    e. **$ passwd idallen-ubuntu**

68. What would be the output of the following command line:
```
echo a b c d | awk '{print $2}'
```
    a. **a b** *b.* **b** *c.* **$2**
    d. no output *e.* **c d**

69. What would be the output of the following command line:
```
echo a b c d | awk '{print $NF}'
```
    a. **a b c d** *b.* **$NF** *c.* **d**
    d. no output *e.* **4**

70. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-wx-w-rwx 2 pat bg1 60 Jan 1 1:00 foo
-rwxrwxrwx 1 pat ted  0 Jan 1 1:00 foo/bar
```
    a. **bob** can rename the file
    b. **bob** can access and write on the file
    c. **bob** can create a new file in the directory
    d. **bob** can list names in the directory
    e. **pat** can create a new file in the directory

71. Which command line below does not show any lines from inside the file **bat**?
    a. **less bat** *b.* **tail bat** *c.* **more bat**
    d. **head bat** *e.* **ls bat**

72. If **a=123** and **b=456** then what is the output of the following sequence of commands: **if $a = $b ; then echo $a ; fi**
    a. **test: a=123: integer expression expected**
    b. no output
    c. **test: $a: string expression expected**
    d. **bash: 123: command not found**
    e. **123**

73. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-wx----w- 2 pat pgg 60 Jan 1 1:00 foo
-rwxrwxr-x 1 bob bg2  0 Jan 1 1:00 foo/bar
```
    a. **pat** can rename the file
    b. **bob** can create a new file in the directory
    c. **pat** can access and write on the file
    d. **bob** can list names in the directory
    e. **bob** can access and write on the file

74. Which of these statements is true?
    a. you can only remove a file name if the file is writable by you
    b. you can only remove a file name if the file is owned by you
    c. you can change the permissions of any file to which you can write
    d. you may be able to rename a file even if you do not own the file
    e. you can only make links to files owned by you

75. Given my directory **dir** and my file **dir/bar** which permissions allow me to access and append data to the file but not delete the file?
    a. Permissions **400** on directory **dir** and **400** on file **dir/bar**.
    b. Permissions **100** on directory **dir** and **100** on file **dir/bar**.
    c. Permissions **100** on directory **dir** and **200** on file **dir/bar**.
    d. Permissions **600** on directory **dir** and **700** on file **dir/bar**.
    e. Permissions **200** on directory **dir** and **200** on file **dir/bar**.

76. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr-xr-x-w- 2 bob pgg 60 Jan 1 1:00 foo
-r-xrwxr-x 1 bob bg1  0 Jan 1 1:00 foo/bar
```
    a. **pat** can access and write on the file
    b. **bob** can access and write on the file
    c. **bob** can list names in the directory
    d. **pat** can rename the file
    e. **bob** can create a new file in the directory

77. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-w-rw---x 2 bob ted 60 Jan 1 1:00 foo
--w-rwxrwx 1 pat bg1  0 Jan 1 1:00 foo/bar
```
    a. **bob** can list names in the directory
    b. **bob** can access and write on the file
    c. **pat** can rename the file
    d. **bob** can create a new file in the directory
    e. **pat** can access and write on the file

78. What command displays the groups you are in?
    a. **grouprint** *b.* **gpasswd** *c.* **groups**
    d. **lstgroups** *e.* **mkgroups**

79. Which of these outputs an error message on Standard Error?
    a. `echo 1>$2 'error'` *b.* `echo 1>2 'error'`
    c. `echo 2>$1 'error'` *d.* `echo 2>&1 'error'`
    e. `echo 1>&2 'error'`

80. Dereference the following symlink **bar** into its equivalent absolute path:
```
ln -s ../b/../b/../../foo /tmp/a/b/bar
```
    a. `/tmp/a/b/bar` *b.* `/tmp/b/bar` *c.* `/tmp/a/foo`
    d. `/tmp/b/foo` *e.* `/tmp/foo`

81. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d---rwx--x 2 pat pgg 60 Jan 1 1:00 foo
--w----rwx 1 bob bg1  0 Jan 1 1:00 foo/bar
```
    a. **bob** can list names in the directory
    b. **pat** can rename the file
    c. **bob** can create a new file in the directory
    d. **bob** can access and write on the file
    e. **pat** can access and write on the file

82. Which of the following commands would result in an error?
    a. `[ 3 -eq 4 ]` *b.* `[ a -eq 4 ]` *c.* `[ 3 = 4 ]`
    d. `[ a = 4 ]` *e.* `[ a != 4 ]`

83. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-wxr-xrw- 2 bob pgg 60 Jan 1 1:00 foo
-r-xrwxr-x 1 bob bg1  0 Jan 1 1:00 foo/bar
```
    a. **bob** can create a new file in the directory
    b. **pat** can rename the file
    c. **pat** can access and write on the file
    d. **bob** can access and write on the file
    e. **bob** can list names in the directory

84. Given the following, can user **bird** in group **sesame** append to **./foo**?
```
dr-xr--r-x 2 root sesame 4096 Oct 7 14:00 .
-rw-rw-r-- 1 bird sesame 123  Oct 4 14:05 foo
```
    a. No, because the directory is not accessible to **bird**
    b. No, because **bird** has no write permission on the directory
    c. Yes, because **bird** has write permissions on **foo**
    d. Yes; permissions don't apply because **bird** owns **foo**
    e. No, because execute permissions are not set for **bird** on **foo**

85. What command manipulates your personal list of repeated scheduled commands:
    a. `crontab` *b.* `showall` *c.* `psmine`
    d. `dmesg` *e.* `ps lxww`

86. When a personal **crontab** job runs, the current working directory is set to:
    a. the directory with the name **/root**
    b. the HOME directory of the user who created the job
    c. the directory with the name **/home**
    d. the current directory that was in use when the **crontab** job was created
    e. the system ROOT directory

87. In a directory containing one file named **dog**, what is the output on your screen after this command line: `2>/dev/null ls nosuchfile`
    a. no output
    b. `dog`
    c. `ls: nosuchfile: No such file or directory`
    d. `nosuchfile`
    e. `bash: 2>/dev/null: command not found`

88. To send a **KILL** signal to a process with process ID *PID*, which of the following commands would you use?
    a. `kill -KILL` *PID* *b.* `send -KILL` *PID*
    c. `send` *PID* `KILL` *d.* `signal -KILL` *PID*
    e. `kill` *PID* `KILL`

89. In an empty directory, what permissions are on file **???** after these commands:
```
touch ??? *** ; chmod 111 *
chmod 222 ? ; chmod 444 '*'
```
    a. `rw-rw-rw-` *b.* `--x--x--x` *c.* `-w--w--w-`
    d. `-wx-wx-wx` *e.* `r--r--r--`

90. Which command sequence correctly compares the two numbers and prints **OK**?
    a. `if ( ! 4 < 3 ) ; then echo OK ; fi`
    b. `if [ 4 > 3 ] ; then echo OK ; fi`
    c. `if [ 4 -ge 3 ] ; then echo OK ; fi`
    d. `if [ ! 4 -gt 3 ] ; then echo OK ; fi`
    e. `if ( 3 < 4 ) ; then echo OK ; fi`

91. What is the output on your screen of the following command sequence:
```
a=1 ; b=2 ; test $b -ge $a ; echo $?
```
    a. no output on screen
    b. `test: $b: integer expression expected`
    c. `1`
    d. `0`
    e. the number 1 or 0 followed by another 1 or 0 on a new line

92. If I mount **sda1** on **/one** and **sda2** on **/two,** how can I link the existing file **/one/foo** to the new pathname **/two/bar**?
a. **ln -s /one/foo /two/bar**
b. **ln -s /two/bar /one/foo**
c. **ln /one/bar /two/foo**
d. **ln /two/bar /one/foo**
e. **ln /one/foo /two/bar**

93. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-wxrwx-w- 2 pat ted 60 Jan 1 1:00 foo
-r-xr-xrwx 1 pat bg1  0 Jan 1 1:00 foo/bar
```
a. **bob** can access and write on the file
b. **bob** can create a new file in the directory
c. **bob** can list names in the directory
d. **pat** can access and write on the file
e. **pat** can rename the file

94. What command would you use to see the command that **at** job number **2** will run?
a. **at -l 2** *b.* **at -v 2** *c.* **at -m 2**
d. **atq 2** *e.* **at -c 2**

95. If a shell script named **foo** contains the line:
```
if [ '$3' = "$1" ] ; then echo SAME ; fi
```
then which of the following command lines will produce **SAME** as output?
a. **./foo '$3' bar** *b.* **./foo $3 $3**
c. **./foo bar bar** *d.* **./foo "$1" '$3'**
e. **./foo "bar" 'bar'**

96. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-wx--x--x 2 bob ted 60 Jan 1 1:00 foo
-r-xr-xrwx 1 pat bg2  0 Jan 1 1:00 foo/bar
```
a. **pat** can access and write on the file
b. **bob** can list names in the directory
c. **pat** can rename the file
d. **bob** can create a new file in the directory
e. **bob** can access and write on the file

97. What would the following command do: **at 2pm**
a. issue an error message
b. run the user's **crontab** jobs every day at 2pm
c. read commands from stdin to be run once at 2pm
d. run the user's **crontab** jobs at 2pm
e. read commands from stdin to be run every day at 2pm

98. Which line is from the Standard Script Header in this course?
a. **PATH=/bin:/user/bin** *b.* **PATH=/bin:user/bin**
c. **PATH=/bin:/urs/bin** *d.* **PATH=/bin:ur/bin**
e. **PATH=/bin:/usr/bin**

99. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr-xrwxrw- 2 pat pgg 60 Jan 1 1:00 foo
--w----r-x 1 bob bg1  0 Jan 1 1:00 foo/bar
```
a. **bob** can access and write on the file
b. **bob** can create a new file in the directory
c. **pat** can rename the file
d. **bob** can list names in the directory
e. **pat** can access and write on the file

100. Given my directory **dir** and my file **dir/bar** which permissions allow me to access and append data to the file but not delete the file?
a. Permissions **100** on directory **dir** and **100** on file **dir/bar**.
b. Permissions **500** on directory **dir** and **600** on file **dir/bar**.
c. Permissions **300** on directory **dir** and **200** on file **dir/bar**.
d. Permissions **400** on directory **dir** and **400** on file **dir/bar**.
e. Permissions **600** on directory **dir** and **700** on file **dir/bar**.

101. If a shell script **myscript.sh** is called this way:
```
./myscript.sh a b c
```
and the first line inside the script below the script header is
```
echo "$#$1" ; shift
```
what is the output of that line?
a. **3a** *b.* **2a** *c.* **2b** *d.* **4c** *e.* **3b**

102. Given the following, can user **bird** in group **sesame** rename **./foo** to **bar**?
```
d----wx--- 2 root sesame 4096 Oct 7 14:00 .
---------- 1 bird sesame 123  Oct 4 14:05 foo
```
a. No, because **bird** cannot read the directory
b. Yes, because **bird**'s group matches the group writable directory
c. Yes; permissions don't apply because **bird** owns **foo**
d. No, because the directory has no permissions for other users
e. No, because **bird** has no permissions on **foo**

103. Which command counts the number of Unix permission groups you are in?
a. **wc groups** *b.* **umask | wc**
c. **id | wc** *d.* **groups | wc**
e. **echo groups | wc**

104. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr-x-wx--x 2 bob ted 60 Jan 1 1:00 foo
-r-xr-xrwx 1 pat bg1  0 Jan 1 1:00 foo/bar
```
a. **bob** can list names in the directory
b. **pat** can rename the file
c. **pat** can access and write on the file
d. **bob** can access and write on the file
e. **bob** can create a new file in the directory

105. In a shell **case** structure, the **case** segment that will GLOB match the text **a**, **b**, or **c**, is coded as
   a. **a/b/c)** *b.* **a\b\c)** *c.* **a|b|c)**
   d. **a,b,c)** *e.* **a:b:c)**

106. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr-xrwx-wx 2 pat ted 60 Jan 1 1:00 foo
-r-xr-xrwx 1 pat bg2  0 Jan 1 1:00 foo/bar
```
   a. **bob** can access and write on the file
   b. **pat** can create a new file in the directory
   c. **bob** can list names in the directory
   d. **bob** can rename the file
   e. **pat** can access and write on the file

107. Which command line displays all the non-hidden names in the current directory that contain the case-insensitive word **hi** (and no other names)?
   a. **echo *(H,h,I,i)*** *b.* **echo *[Hh][Ii]***
   c. **echo *[hiHI]*** *d.* **echo ?[HhIi]?**
   e. **echo ?[HhIiHhIi]?**

108. If **archive.tar.gz** is a compressed tar archive, which command could you run to produce a listing of its contents without extracting it?
   a. **tar -tgz archive**
   b. **tar -tzf archive**
   c. **tar -tgz archive.tar.gz**
   d. **tar -xzf archive.tar.gz**
   e. **tar -tzf archive.tar.gz**

109. The **minimum** permissions you need to read a file **foo** in directory **a** are:
   a. **x** on **a**, **r** on **foo** *b.* **wx** on **a**, **w** on **foo**
   c. **rwx** on **a**, none on **foo** *d.* **rwx** on **a**, **rw** on **foo**
   e. **wx** on **a**, none on **foo**

110. What value **umask** gives a new file permissions **r--r-----**?
   a. **110** *b.* **440** *c.* **220** *d.* **446** *e.* **337**

111. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr---wx--x 2 bob ted 60 Jan 1 1:00 foo
--w--w-r-x 1 bob bg2  0 Jan 1 1:00 foo/bar
```
   a. **bob** can create a new file in the directory
   b. **pat** can rename the file
   c. **bob** can list names in the directory
   d. **bob** can access and write on the file
   e. **pat** can access and write on the file

112. The **minimum** permissions you need to link a file **foo** from directory **a** to directory **b** are:
   a. **wx** on **a**, **wx** on **b**, **w** on **foo**
   b. **x** on **a**, **wx** on **b**, none on **foo**
   c. **rwx** on **a**, **wx** on **b**, **rw** on **foo**
   d. **wx** on **a**, **wx** on **b**, **r** on **foo**
   e. **rwx** on **a**, **wx** on **b**, none on **foo**

113. In an empty directory, what is output on your screen by:
```
mkdir -p a/b/c 1/2/3 ; mv a/b 1/2 ; find . -name c
```
   a. **./1/a/b** *b.* **./1/2/b/c** *c.* **./1/2/c**
   d. **./1/2/a/b** *e.* **./a/b/c**

114. Given the following, can user **bird** in group **sesame** copy **./foo** to **bar**?
```
drwxr-xrwx 2 root sesame 4096 Oct 7 14:00 .
-r-xr-xr-x 1 bird sesame 123  Oct 4 14:05 foo
```
   a. Yes, because **bird** has read permissions on **foo**
   b. No, because the directory has no write permissions for **bird**
   c. Yes; permissions don't apply because **bird** owns **foo**
   d. No, because the directory is not accessible to **bird**
   e. No, because **foo** has no write permissions for **bird**

115. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d--xr-x-w- 2 bob pgg 60 Jan 1 1:00 foo
--w----r-x 1 bob bg2  0 Jan 1 1:00 foo/bar
```
   a. **bob** can list names in the directory
   b. **pat** can rename the file
   c. **pat** can access and write on the file
   d. **bob** can access and write on the file
   e. **bob** can create a new file in the directory

116. In an empty directory, what is output on your screen by:
```
mkdir -p a/b/c 1/2/3 ; mv a/b 1/2/3 ; find . -name c
```
   a. **./a/b/c** *b.* **./1/2/3/c** *c.* **./1/2/3/a/b**
   d. **./1/2/a/b** *e.* **./1/2/3/b/c**

117. Which expands to the exit status of the previous command?
   a. **"$*"** *b.* **"$@"** *c.* **"$#"** *d.* **"$0"** *e.* **"$?"**

118. Given the following, can user **bird** in group **sesame** remove **./foo**?
```
drwxr-xrwx 2 root sesame 4096 Oct 7 14:00 .
-rwxrwxrwx 1 bird sesame 123  Oct 4 14:05 foo
```
   a. Yes, because **bird** has full permissions on **foo**
   b. Yes, because **bird** matches the writable other permissions
   c. Yes; permissions don't apply because **bird** owns **foo**
   d. No, because the directory is not accessible to **bird**
   e. No, because **bird** has no write permission on the directory

119. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-w---xr-x 2 pat ted 60 Jan 1 1:00 foo
-rwxr-xrwx 1 pat bg2  0 Jan 1 1:00 foo/bar
```
a. **bob** can list names in the directory
b. **bob** can rename the file
c. **bob** can access and write on the file
d. **pat** can access and write on the file
e. **bob** can create a new file in the directory

120. Given my directory **dir** and my file **dir/bar** which permissions allow me to delete the file from the directory, but not append data to the file?
a. Permissions **600** on directory **dir** and **500** on file **dir/bar**.
b. Permissions **500** on directory **dir** and **500** on file **dir/bar**.
c. Permissions **300** on directory **dir** and **100** on file **dir/bar**.
d. Permissions **600** on directory **dir** and **300** on file **dir/bar**.
e. Permissions **700** on directory **dir** and **200** on file **dir/bar**.

121. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr-xrw-rwx 2 pat bg1 60 Jan 1 1:00 foo
-rwxrwxrwx 1 pat ted  0 Jan 1 1:00 foo/bar
```
a. **pat** can rename the file
b. **bob** can list names in the directory
c. **pat** can create a new file in the directory
d. **bob** can access and write on the file
e. **bob** can rename the file

122. Inside a shell script, which expands to the name of the script itself?
a. **"$*"** *b.* **"$@"** *c.* **"$0"** *d.* **"$#"** *e.* **"$?"**

123. In an empty directory, what permissions are on file **???** after these commands:
```
touch ??? *** ; chmod 111 *
chmod 222 ??? ; chmod 444 '***'
```
a. **rw-rw-rw-** *b.* **--x--x--x** *c.* **-w--w--w-**
d. **r--r--r--** *e.* **-wx-wx-wx**

124. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr-x-wx--- 2 pat bg1 60 Jan 1 1:00 foo
-rwxrwxr-x 1 pat ted  0 Jan 1 1:00 foo/bar
```
a. **pat** can rename the file
b. **pat** can create a new file in the directory
c. **bob** can create a new file in the directory
d. **bob** can list names in the directory
e. **bob** can access and write on the file

125. What command terminates processes based on their name (not safe!):
a. **crontab** *b.* **killall** *c.* **dmesg**
d. **ps lxww** *e.* **kill**

126. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d--xr----x 2 bob ted 60 Jan 1 1:00 foo
--w--w-r-x 1 bob bg1  0 Jan 1 1:00 foo/bar
```
a. **bob** can access and write on the file
b. **bob** can list names in the directory
c. **pat** can access and write on the file
d. **bob** can create a new file in the directory
e. **pat** can rename the file

127. Given the following, can user **bird** in group **sesame** append to **./foo**?
```
dr-xr-xr-x 2 root sesame 4096 Oct 7 14:00 .
-rw-r-xr-x 1 bird sesame 123  Oct 4 14:05 foo
```
a. Yes; permissions don't apply because **bird** owns **foo**
b. No, because execute permissions are not set for **bird** on **foo**
c. No, because **bird** has no write permission on the directory
d. No, because the directory is not accessible to **bird**
e. Yes, because **bird** has write permissions on **foo**

128. Which command displays all processes in a full wide listing?
a. **ps zxvf** *b.* **ps -all -wide**
c. **ps -full** *d.* **ps -any -wide**
e. **ps laxww**

129. What is the output on your screen of the following sequence of commands:
```
a=4 ; b=4 ; [ $a -le $b ] ; echo $?
```
a. **1**
b. **test: $a: integer expression expected**
c. no output
d. **0**
e. the number 1 or 0 followed by another 1 or 0 on a new line

130. The signal sent to a foreground process by typing the **[Ctrl-C]** key is:
a. **SIGTERM** *b.* **SIGINT** *c.* **SIGKILL**
d. **SIGHUP** *e.* **SIGSTOP**

131. What value **umask** gives a new directory permissions **rw--w---x**?
a. **211** *b.* **156** *c.* **421** *d.* **621** *e.* **432**

132. Given my directory **dir** and my file **dir/bar** which permissions allow me to access and append data to the file but not delete the file?
a. Permissions **200** on directory **dir** and **200** on file **dir/bar**.
b. Permissions **500** on directory **dir** and **100** on file **dir/bar**.
c. Permissions **500** on directory **dir** and **200** on file **dir/bar**.
d. Permissions **400** on directory **dir** and **400** on file **dir/bar**.
e. Permissions **600** on directory **dir** and **700** on file **dir/bar**.

133. A shell script named **bar** is executed as follows:
```
./bar "a b" "c d e" f
```
Inside the script is the line: **echo "$3"**
What is the output on your screen from this line?
a. **c d e** *b.* **f** *c.* **a b**
d. **$3** *e.* **"f"**

134. What value to **chmod** would change the permissions on a file to **rw-r--r--**?
a. **311** *b.* **344** *c.* **644** *d.* **244** *e.* **211**

135. Which of these safely tests for a null (empty) first argument?
a. **if ["$1" -eq '']**
b. **if [$1 = ""]**
c. **if ["$1" = '']**
d. **if [$1 = '']**
e. **if ["$1" -eq '/dev/null']**

136. The *difference* between the system (**root**) crontab and all the user (personal) crontabs is:
a. the system crontab has the date and time in it
b. the system crontab also has the userid in it
c. the personal crontab also has the userid in it
d. the personal crontab has the date and time in it
e. the personal crontab only runs commands once

137. To bring a background shell job into the foreground, type:
a. **fg** *b.* **bg** *c.* **kill %1**
d. **[Ctrl-D]** *e.* **[Ctrl-Z]**

138. When a user named **bob** runs a command in a **setuid** executable file owned by **foo**, in a directory owned by **root**, the file executes with the permissions of:
a. **bob** *b.* **foo** *c.* **root and bob**
d. **root and foo** *e.* **root**

139. If a shell script **myscript.sh** is called this way:
```
./myscript.sh a b c
```
and the first line inside the script below the script header is
```
shift ; echo "$#$1"
```
what is the output of that line?
a. **3b** *b.* **2b** *c.* **4c** *d.* **2a** *e.* **3a**

140. Which of the following could you use as options for the **tar** command to extract a **gzip**-compressed archive?
a. **-tgz** *b.* **egf** *c.* **xzf** *d.* **ezf** *e.* **-czf**

141. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d--x-wx--- 2 bob pgg 60 Jan 1 1:00 foo
-r-x-w-r-x 1 bob bg1  0 Jan 1 1:00 foo/bar
```
a. **bob** can list names in the directory
b. **pat** can rename the file
c. **pat** can access and write on the file
d. **bob** can access and write on the file
e. **bob** can create a new file in the directory

142. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d-wx---rw- 2 bob ted 60 Jan 1 1:00 foo
----rwxrwx 1 bob bg2  0 Jan 1 1:00 foo/bar
```
a. **pat** can rename the file
b. **bob** can access and write on the file
c. **pat** can access and write on the file
d. **bob** can list names in the directory
e. **bob** can create a new file in the directory

143. Which command line below does not show any lines from inside the file **out**?
a. **wc out** *b.* **head out** *c.* **sort out**
d. **more out** *e.* **tail out**

144. Given the following, can user **bird** in group **sesame** append to **./foo**?
```
dr-xr-xr-x 2 root sesame 4096 Oct 7 14:00 .
-r-xrwxrwx 1 bird sesame 123  Oct 4 14:05 foo
```
a. Yes; permissions don't apply because **bird** owns **foo**
b. No, because **bird** has no write permissions on **foo**
c. No, because execute permissions are not set for **bird** on **foo**
d. No, because the directory is not accessible to **bird**
e. No, because **bird** has no write permission on the directory

145. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr-x----wx 2 pat ted 60 Jan 1 1:00 foo
-r-xr-xrwx 1 pat bg1  0 Jan 1 1:00 foo/bar
```
a. **bob** can access and write on the file
b. **bob** can create a new file in the directory
c. **pat** can access and write on the file
d. **bob** can list names in the directory
e. **pat** can rename the file

146. Given the following, can user **bird** in group **sesame** copy **./foo** to **bar**?
```
drwx-wx--x 2 root sesame 4096 Oct 7 14:00 .
--wxrwxrwx 1 bird sesame 123  Oct 4 14:05 foo
```
a. No, because the directory has no write permissions for **bird**
b. Yes, because **bird** has write permissions on **foo**
c. Yes; permissions don't apply because **bird** owns **foo**
d. No, because **foo** has no read permissions for **bird**
e. No, because the directory is not readable by **bird**

147. Dereference the following symlink **bar** into its equivalent absolute path:
```
ln -s ../b/../../a/./foo /tmp/a/b/bar
```
a. **/tmp/foo** *b.* **/tmp/a/b/bar** *c.* **/tmp/a/foo**
d. **/tmp/b/bar** *e.* **/tmp/b/foo**

148. Other than **root**, who can change the permissions of the following directory?
```
dr-xrwxrwx 17 foo bar 4096 Apr 15 16:40 .
```
a. user **foo** and any user in group **bar**
b. only user **foo**
c. only users in group **bar**
d. only **root** can change the permissions
e. anyone except user **foo**

149. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d--xrwx-wx 2 bob ted 60 Jan 1 1:00 foo
-r-x-w-r-x 1 bob bg2  0 Jan 1 1:00 foo/bar
```
a. **pat** can access and write on the file
b. **bob** can create a new file in the directory
c. **bob** can list names in the directory
d. **bob** can access and write on the file
e. **pat** can rename the file

150. If the current directory contains 10 visible files and 5 visible sub-directories, what is the output on your screen of this command: **ls -d */.**
a. an error message because ***/.** does not exist
b. 5 directory names
c. no output
d. 15 pathnames
e. ***/.**

151. In an empty directory, what is output on your screen by:
```
mkdir -p a/b/c 1/2/3 ; mv a 1/2 ; find . -name c
```
a. **./1/2/3/a/b/c** *b.* **./1/a**
c. **./1/2/a/b/c** *d.* **./1/2/a**
e. **./1/2/3/a/b**

152. Which of the following options for **bash** or **sh** might be useful for debugging a shell script?
a. **-x** *b.* **-c** *c.* **-l** *d.* **-z** *e.* **-r**

153. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
drw------x 2 pat ted 60 Jan 1 1:00 foo
--w--w-r-x 1 pat bg1  0 Jan 1 1:00 foo/bar
```
a. **pat** can access and write on the file
b. **bob** can access and write on the file
c. **bob** can list names in the directory
d. **bob** can create a new file in the directory
e. **bob** can rename the file

154. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr---wx--- 2 pat bg2 60 Jan 1 1:00 foo
-rw-rw-r-x 1 pat ted  0 Jan 1 1:00 foo/bar
```
a. **pat** can create a new file in the directory
b. **bob** can list names in the directory
c. **bob** can rename the file
d. **pat** can rename the file
e. **bob** can access and write on the file

155. Inside a shell script, which expands to the number of script arguments?
a. **"$#"** *b.* **"$0"** *c.* **"$@"** *d.* **"$?"** *e.* **"$*"**

156. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d--xrwx--x 2 bob ted 60 Jan 1 1:00 foo
----rw--w- 1 bob bg1  0 Jan 1 1:00 foo/bar
```
a. **bob** can create a new file in the directory
b. **bob** can access and write on the file
c. **pat** can access and write on the file
d. **bob** can list names in the directory
e. **pat** can rename the file

157. What is the output on your screen of the following sequence of commands:
```
x=ok ; y=ok ; [ x = y ]
```
a. **test: x: integer expression expected**
b. **bash: x: command not found**
c. no output on screen
d. **1**
e. **0**

158. What command changes a user's password?
a. **chsh** *b.* **password** *c.* **chpasswd**
d. **passwd** *e.* **mkpasswd**

159. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
drw-r-xrwx 2 pat bg1 60 Jan 1 1:00 foo
-rwxrwxr-x 1 pat ted  0 Jan 1 1:00 foo/bar
```
a. **bob** can rename the file
b. **bob** can access and write on the file
c. **bob** can list names in the directory
d. **pat** can rename the file
e. **pat** can create a new file in the directory

160. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
drw---x--- 2 pat bg2 60 Jan 1 1:00 foo
-r------w- 1 pat ted  0 Jan 1 1:00 foo/bar
```
a. **pat** can rename the file
b. **bob** can access and write on the file
c. **pat** can create a new file in the directory
d. **bob** can rename the file
e. **bob** can list names in the directory

161. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
d--x--xrw- 2 bob pgg 60 Jan 1 1:00 foo
-r-xrwx-w- 1 bob bg2  0 Jan 1 1:00 foo/bar
```
a. **pat** can access and write on the file
b. **bob** can list names in the directory
c. **bob** can create a new file in the directory
d. **pat** can rename the file
e. **bob** can access and write on the file

162. Given this successful command line (note the dot argument):
```
cd /home/foo ; mkdir bar ; cd bar ; chmod a-x .
```
Which of the following subsequent commands will execute without any "permission denied" errors?
a. **ls /home/foo/bar/.** *b.* **ls /home/foo/bar**
c. **ls .** *d.* **ls ..**
e. **ls /home/foo/bar/..**

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

164. Given the following, can user **bird** in group **sesame** copy **./foo** to **bar**?
```
drwxrw-r-x 2 root sesame 4096 Oct 7 14:00 .
-rwx-wx-wx 1 bird sesame 123  Oct 4 14:05 foo
```
a. Yes, because **bird** has write permissions on **foo**
b. No, because **foo** has no read permissions for **bird**
c. No, because the directory has no write permissions for others
d. Yes; permissions don't apply because **bird** owns **foo**
e. No, because the directory is not accessible to **bird**

165. Given my directory **dir** and my file **dir/bar** which permissions allow me to delete the file from the directory, but not append data to the file?
a. Permissions **500** on directory **dir** and **500** on file **dir/bar**.
b. Permissions **600** on directory **dir** and **300** on file **dir/bar**.
c. Permissions **600** on directory **dir** and **500** on file **dir/bar**.
d. Permissions **700** on directory **dir** and **200** on file **dir/bar**.
e. Permissions **700** on directory **dir** and **500** on file **dir/bar**.

166. A Unix/Linux "tarball" is:
a. a single-file that contains individual uncompressed files
b. a multi-file directory containing individual compressed files
c. a single compressed file containing one uncompressed file
d. a multi-file directory containing individual uncompressed files
e. a single-file that contains individual compressed files

167. What permissions are given to **newdir** after this command line:
```
umask 156 ; mkdir newdir
```
a. **r-x-w-rw-** *b.* **r-x--x---** *c.* **--xr-xrw-**
d. **rw--w----** *e.* **rw--w---x**

168. The **minimum** permissions you need to delete a file **foo** from directory **a** are:
a. **wx** on **a**, **r** on **foo** *b.* **wx** on **a**, **w** on **foo**
c. **rwx** on **a**, **rw** on **foo** *d.* **rwx** on **a**, none on **foo**
e. **wx** on **a**, none on **foo**

169. Under what directory are system log files usually stored?
a. **/usr/bin** *b.* **/etc/log** *c.* **/log/var**
d. **/var/log** *e.* **/bin/**

170. If **browser=lynx** then which one of the following **case** patterns will match this statement: **case "$browser" in**
a. **?lynx?) echo yes ;;**
b. **(*ynx echo yes ;;**
c. **[lynx] | [LYNX]) echo yes ;;**
d. **l?n?) echo yes ;;**
e. **@) echo yes ;;**

171. What value **umask** gives a new file permissions **r--r-----**?
a. **220** *b.* **440** *c.* **226** *d.* **110** *e.* **446**

172. Which of the following commands would result in an error?
a. **[3 -eq 4]** *b.* **[a = 4]** *c.* **[3 -e 3]**
d. **[a != 4]** *e.* **[3 = f]**

173. The **minimum** permissions you need to move a file **foo** from directory **a** to directory **b** are:
a. **wx** on **a**, **wx** on **b**, **w** on **foo**
b. **wx** on **a**, **wx** on **b**, **r** on **foo**
c. **rwx** on **a**, **wx** on **b**, none on **foo**
d. **rwx** on **a**, **wx** on **b**, **rw** on **foo**
e. **wx** on **a**, **wx** on **b**, none on **foo**

174. What command displays the kernel ring buffer of log messages:
a. **dmesg** *b.* **crontab** *c.* **ps lxww**
d. **psmine** *e.* **showall**

175. If **guru=linus** then which one of the following **case** patterns will match this statement: **case "$guru" in**
   a. ***) echo yes ;;**
   b. **lin?) echo yes ;;**
   c. **(*nus echo yes ;;**
   d. **[linus] | [LINUS]) echo yes ;;**
   e. **"linu?") echo yes ;;**

176. Which command line makes a directory **dir** into which anyone can put a file, but in which nobody can see the names of the files that are there?
   a. **mkdir dir ; chmod 777 .**
   b. **mkdir dir ; cd dir ; chmod go+wx .**
   c. **mkdir dir ; chmod 777 dir**
   d. **mkdir dir ; cd dir ; chmod go-x .**
   e. **mkdir dir ; chmod 333 dir**

177. When an **at** job runs, the current working directory is set to:
   a. the current directory that was in use when the **at** job was created
   b. the HOME directory of the user who created the job
   c. the directory with the name **/root**
   d. the directory with the name **/home**
   e. the system ROOT directory

178. Under what directory are system configuration files usually stored?
   a. **/usr/bin** *b.* **/log/var/** *c.* **/bin/**
   d. **/etc** *e.* **/var/log/**

179. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
```
dr--r-x-w- 2 bob pgg 60 Jan 1 1:00 foo
-rwxrwxr-x 1 bob bg2  0 Jan 1 1:00 foo/bar
```
   a. **bob** can list names in the directory
   b. **bob** can access and write on the file
   c. **pat** can access and write on the file
   d. **pat** can rename the file
   e. **bob** can create a new file in the directory

180. How does system logging work under Unix/Linux?
   a. processes write log files into each user's **$HOME** directory
   b. processes send messages to a central **rsyslog** program that writes log files
   c. processes send messages to the **init** process that inherits orphan processes
   d. processes write log entries directly into the system log directory
   e. processes copy logs from your **$HOME** directory to the **/var/spool** directory

181. The password **:x:** in **/etc/passwd** means:
   a. the encrypted password is **"x"**
   b. the unencrypted password is stored in the group file
   c. the password is locked
   d. the encrypted password is stored in the shadow file
   e. the account is locked

182. **Did you read all the words of the test instructions on page one?**
   a. **123** *b.* **231** *c.* **321** *d.* **132** *e.* **312**