PRINT Name: ______________________________

One-Answer Multiple Choice **188 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 last question about reading these test instructions is: **`123`**

191. Answer **`191`** is ○ ○ ○ D ○
192. Answer **`192`** is A ○ ○ ○ ○
193. Answer **`193`** is ○ B ○ ○ ○
194. Answer **`194`** is ○ B ○ ○ ○
195. Answer **`195`** is A ○ ○ ○ ○
196. Answer **`196`** is ○ ○ ○ D ○

Your **Test Version** is:

D A B B 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. 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. **`dog`**
   d. **`bash: 1>/dev/null: command not found`**
   e. **`ls: *: No such file or directory`**
2. Can three different files have the same inode number on three different file systems?
   a. no: inode numbers only apply to directories, not files
   b. no: you can't have inode numbers on three file systems
   c. no: inode numbers are unique across all file systems
   d. yes: inode numbers are only unique inside a file system
   e. yes: if the files are all names for the same inode
3. When a personal **`crontab`** job runs, the current working directory is set to:
   a. the directory with the name **`/home`**
   b. the system ROOT directory
   c. the directory with the name **`/root`**
   d. the HOME directory of the user who created the job
   e. the current directory that was in use when the **`crontab`** job was created

4. What is the output on your screen of the following sequence of commands:
   `x=pig ; [ -z $x ] ; echo $?`
   a. **`0`**
   b. no output
   c. **`test: $x: integer expression expected`**
   d. the number 0 or 1 followed by another 0 or 1 on a new line
   e. **`1`**
5. 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. **`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
6. A **`crontab`** entry of **`0 6 * * * /sbin/somescript`** would run **`somescript`** when and how often?
   a. at 12:06am every day
   b. at 12:06am every business day
   c. at 6:00am every business day
   d. at 12:06am every business day and Saturday
   e. at 6:00am every day
7. 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. **`bob`** can rename the file
   b. **`pat`** 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
8. 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 -s /a/foo /b/new`**
   b. **`ln /b/new /a/foo`**
   c. **`ln /a/new /b/foo`**
   d. **`ln -s /b/new /a/foo`**
   e. **`ln /a/foo /b/new`**
9. 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 create a new file in the directory
   c. **`bob`** can list names in the directory
   d. **`pat`** can rename the file
   e. **`pat`** can access and write on the file

10. 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. 7 iterations
b. 8 iterations
c. 6 iterations
d. 9 iterations
e. 1 iteration

11. Given my directory containing a file, which octal permissions allow me to access and append data to the file but not delete the file?

a. Directory: **600** File: **700**
b. Directory: **400** File: **400**
c. Directory: **500** File: **100**
d. Directory: **500** File: **200**
e. Directory: **200** File: **200**

12. 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. **bob** can create a new file in the directory
b. **bob** can list names in the directory
c. **bob** can access and write on the file
d. **pat** can access and write on the file
e. **pat** can rename the file

13. 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 ..**
c. **ls /home/foo/bar/..**
d. **ls /home/foo/bar**
e. **ls .**

14. The **minimum** permissions you need to append to a file **foo** in directory **a** are:

a. **rwx** on **a**, none on **foo**
b. **rwx** on **a**, **rw** on **foo**
c. **x** on **a**, **w** on **foo**
d. **wx** on **a**, none on **foo**
e. **wx** on **a**, **w** on **foo**

15. 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. **4c**
b. **3a**
c. **3b**
d. **2b**
e. **2a**

16. Dereference the following symlink **bar** into its equivalent absolute path:

```
ln -s ../b/../../a/../foo /tmp/a/b/bar
```

a. **/tmp/b/bar**
b. **/tmp/foo**
c. **/tmp/a/b/bar**
d. **/tmp/b/foo**
e. **/tmp/a/foo**

17. 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. **$3**
b. **f**
c. **c d e**
d. **a b**
e. **"f"**

18. What command terminates processes based on their name (not safe!):

a. **dmesg**
b. **crontab**
c. **killall**
d. **ps lxww**
e. **kill**

19. What command manipulates your personal list of repeated scheduled commands:

a. **showall**
b. **ps lxww**
c. **psmine**
d. **dmesg**
e. **crontab**

20. In an empty directory, what permissions are on file **???** after these commands:

```
touch ??? *** ; chmod 111 *
chmod 222 ??? ; chmod 444 '***'
```

a. **rw-rw-rw-**
b. **r--r--r--**
c. **-wx-wx-wx**
d. **--x--x--x**
e. **-w--w--w-**

21. 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. **root and bob**
b. **root**
c. **bob**
d. **root and foo**
e. **foo**

22. 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 $animal ate**
b. **the 'animal' ate**
c. **the 'pig' ate**
d. **the '$animal' ate**
e. **the 'dog' ate**

23. 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. **bob** can list names in the directory
c. **pat** can access and write on the file
d. **bob** can rename the file
e. **pat** can create a new file in the directory

24. The octal mode of a directory that allows the user to create new files in it, but not to list any names in it:

a. **100**
b. **400**
c. **300**
d. **200**
e. **500**

25. 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.* `./1/2/a/b/c`
c. `./1/2/3/a/b` *d.* `./1/2/a`
e. `./1/a`

26. 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 is not readable by **bird**
b. Yes, because **bird** has write permissions on **foo**
c. Yes; permissions don't apply because **bird** owns **foo**
d. No, because the directory has no write permissions for **bird**
e. No, because **foo** has no read permissions for **bird**

27. 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. `./1/2/3/b/c` *b.* `./a/b/c` *c.* `./1/2/3/c`
d. `./1/2/a/b` *e.* `./1/2/3/a/b`

28. The **minimum** permissions you need to read a file **foo** in directory **a** are:

a. **wx** on **a**, **w** on **foo** *b.* **rwx** on **a**, none on **foo**
c. **x** on **a**, **r** on **foo** *d.* **wx** on **a**, none on **foo**
e. **rwx** on **a**, **rw** on **foo**

29. What value **umask** gives a new file permissions **r--r-----**?

a. **440** *b.* **237** *c.* **110** *d.* **446** *e.* **220**

30. 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. **root and foo** *b.* **bob** *c.* **root**
d. **root and bob** *e.* **foo**

31. 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. **bob** can rename the file
b. **pat** can access and write on the file
c. **bob** can list names in the directory
d. **bob** can access and write on the file
e. **bob** can create a new file in the directory

32. 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 access and write on the file
c. **pat** can rename the file
d. **bob** can list names in the directory
e. **bob** can create a new file in the directory

33. 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. **4c** *b.* **2b** *c.* **2a** *d.* **3a** *e.* **3b**

34. Given my directory containing a file, which octal permissions allow me to delete the file from the directory, but not append data to the file?

a. Directory: **500** File: **500** *b.* Directory: **300** File: **400**
c. Directory: **100** File: **500** *d.* Directory: **300** File: **200**
e. Directory: **100** File: **300**

35. Which command counts the number of Unix permission groups you are in?

a. `umask | wc` *b.* `wc groups`
c. `groups | wc` *d.* `id | wc`
e. `echo groups | wc`

36. Which of the following could you use as options for the **tar** command to extract a **gzip**-compressed archive?

a. **-tgz** *b.* **ezf** *c.* **egf** *d.* **-czf** *e.* **xzf**

37. 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 test "" -eq $a ; then date ; fi`
c. `if test "" = "$a" ; then date ; fi`
d. `if [ $a = /dev/null ] ; then date ; fi`
e. `if [ "$a" = * ] ; then date ; fi`

38. 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. No, because **bird** has no write permission on the directory
b. No, because the directory is not accessible to **bird**
c. Yes, because **bird** has write permissions on **foo**
d. No, because execute permissions are not set for **bird** on **foo**
e. Yes; permissions don't apply because **bird** owns **foo**

39. 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. `1234` *b.* `4321` *c.* `1 2 3 4`
d. `1` *e.* `10`

40. What value `umask` gives a new directory permissions `rw--w---x`?

a. `432` *b.* `156` *c.* `211` *d.* `421` *e.* `621`

41. 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. `bob` can list names in the directory
b. `pat` can create a new file in the directory
c. `pat` can rename the file
d. `bob` can access and write on the file
e. `bob` can rename the file

42. 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. the number 0 or 1 followed by another 0 or 1 on a new line
e. `0`

43. In an empty directory, what permissions are on file `???` after these commands:

```
touch ??? *** ; chmod 111 *
chmod 222 ? ; chmod 444 '*'
```

a. `-wx-wx-wx` *b.* `-w--w--w-` *c.* `--x--x--x`
d. `rw-rw-rw-` *e.* `r--r--r--`

44. To send a `KILL` signal to a process with process ID *PID*, which of the following commands would you use?

a. `send` *PID* `KILL` *b.* `send -KILL` *PID*
c. `kill -KILL` *PID* *d.* `kill` *PID* `KILL`
e. `signal -KILL` *PID*

45. 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. `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

46. 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. `pat` can rename 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 access and write on the file

47. 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. `test: $a: string expression expected`
c. no output
d. `123`
e. `bash: 123: command not found`

48. What value to `chmod` would change the permissions on a file to `rw-r--r--`?

a. `644` *b.* `244` *c.* `211` *d.* `344` *e.* `311`

49. To list your personal crontab, type:

a. `cat crontab` *b.* `crontab -l`
c. `/var/log/crontab` *d.* `atq`
e. `/etc/crontab`

50. Which of these statements is true?

a. You only need "`r--`" permission on directory "`foo`" for "`ls -l foo`" to work.
b. To make a hard link to file "`foo`" named "`bar`", file "`foo`" must exist.
c. If you give me write permission on a file owned by you, I can then use `chmod` to change its permissions.
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.

51. 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. **bash: [123: command not found**
b. no output
c. **test: a=123: integer expression expected**
d. **123**
e. **test: $a: string expression expected**

52. Dereference the following symlink **bar** into its equivalent absolute path:

```
ln -s ../b/../../a/./foo /tmp/a/b/bar
```

a. **/tmp/a/b/bar** *b.* **/tmp/foo** *c.* **/tmp/a/foo**
d. **/tmp/b/bar** *e.* **/tmp/b/foo**

53. 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)**

54. 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 333 dir**
b. **mkdir dir ; chmod 777 dir**
c. **mkdir dir ; cd dir ; chmod go+wx .**
d. **mkdir dir ; chmod 777 .**
e. **mkdir dir ; cd dir ; chmod go-x .**

55. 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 access and write on the file
d. **pat** can rename the file
e. **bob** can create a new file in the directory

56. The octal mode of a directory that allows the user to list the names in it, but not to create files or to **cd** into the directory:
a. **200** *b.* **100** *c.* **300** *d.* **500** *e.* **400**

57. 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. **bob** can list names in the directory
b. **bob** can create a new file in the directory
c. **bob** can access and write on the file
d. **pat** can access and write on the file
e. **pat** can rename the file

58. The octal mode of a directory that allows the user to list the names in it, but not to create files or to access any of the files:
a. **200** *b.* **100** *c.* **500** *d.* **400** *e.* **300**

59. 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 -tzf archive**
b. **tar -tgz archive.tar.gz**
c. **tar -tgz archive**
d. **tar -tzf archive.tar.gz**
e. **tar -xzf archive.tar.gz**

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

61. The octal mode of a directory that allows the user to access files if they know their names, but not to list or to change the names:
a. **300** *b.* **200** *c.* **100** *d.* **500** *e.* **400**

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

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

64. What is the output on your screen of the following command sequence:

```
a=1 ; b=2 ; test $b -ge $a ; echo $?
```

a. **test: $b: integer expression expected**
b. **0**
c. no output on screen
d. the number 1 or 0 followed by another 1 or 0 on a new line
e. **1**

65. 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. No, because execute permissions are not set for **bird** 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. Yes, because **bird** has write permissions on **foobar**

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

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

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

69. 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. `pat` can rename the file
b. `bob` can create a new file in the directory
c. `bob` can list names in the directory
d. `bob` can rename the file
e. `bob` can access and write on the file

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

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

72. 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. `bob` can rename the file
b. `pat` can rename the file
c. `pat` can create a new file in the directory
d. `bob` can access and write on the file
e. `bob` can list names in the directory

73. 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 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

74. Given my directory containing a file, which octal permissions allow me to delete the file from the directory, but not append data to the file?
a. Directory: `700` File: `200` *b.* Directory: `600` File: `300`
c. Directory: `500` File: `500` *d.* Directory: `600` File: `500`
e. Directory: `300` File: `100`

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

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

77. 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/foo` *c.* `/tmp/b/bar`
d. `/tmp/a/foo` *e.* `/tmp/foo`

78. 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. `bob` can rename the file
b. `pat` can rename the file
c. `pat` can create a new file in the directory
d. `bob` can access and write on the file
e. `bob` can list names in the directory

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

80. Which of these statements is true?
a. you can only remove a file name if the file is owned by you
b. you can only make links to files 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 remove a file name if the file is writable by you

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

82. 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 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

83. What command would you use to see the command that **at** job number **2** will run?

a. `at -c 2` *b.* `atq 2` *c.* `at -m 2`
d. `at -l 2` *e.* `at -v 2`

84. To change your own account password, type this into the shell prompt:

a. `$ passwd`
b. `$ passwd idallen-ubuntu`
c. `$ passwd .`
d. `$ passwd cst8207`
e. `$ passwd *`

85. When an **at** job runs, the current working directory is set to:

a. the system ROOT directory
b. the directory with the name **/home**
c. the HOME directory of the user who created the job
d. the directory with the name **/root**
e. the current directory that was in use when the **at** job was created

86. What is the output on your screen of the following sequence of commands:

```
    a=4 ; b=4 ; [ $a -le $b ] ; echo $?
```

a. `test: $a: integer expression expected`
b. `0`
c. the number 1 or 0 followed by another 1 or 0 on a new line
d. no output
e. `1`

87. 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. **pat** can rename the file
b. **pat** can access and write on the file
c. **bob** can create a new file in the directory
d. **bob** can access and write on the file
e. **bob** can list names in the directory

88. What permissions are given to **newdir** after this command line:

```
    umask 156 ; mkdir newdir
```

a. `--xr-xrw-` *b.* `rw--w---x` *c.* `r-x--x---`
d. `rw--w----` *e.* `r-x-w-rw-`

89. How does system logging work under Unix/Linux?

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

90. 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 rename the file
b. **bob** can list names in the directory
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

91. 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. **pat** can access and write on the file
c. **pat** can rename the file
d. **bob** can create a new file in the directory
e. **bob** can list names in the directory

92. Which of these outputs an error message on Standard Error?

a. `echo 2>$1 'error'` *b.* `echo 2>&1 'error'`
c. `echo 1>&2 'error'` *d.* `echo 1>2 'error'`
e. `echo 1>$2 'error'`

93. The octal mode of a directory that allows the user to access files and list the names in it, but not to create any new files:

a. `400` *b.* `300` *c.* `500` *d.* `200` *e.* `100`

94. 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 personal crontab has the date and time in it
c. the personal crontab only runs commands once
d. the system crontab also has the userid in it
e. the personal crontab also has the userid in it

95. 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. **bob** can create a new file in the directory
b. **pat** can access and write on the file
c. **pat** can rename the file
d. **bob** can list names in the directory
e. **bob** can access and write on the file

96. To bring a background shell job into the foreground, type:

a. **bg** *b.* **[Ctrl-D]** *c.* **[Ctrl-Z]**
d. **fg** *e.* **kill %1**

97. What value **umask** gives a new file permissions **r--r-----**?

a. **110** *b.* **446** *c.* **220** *d.* **337** *e.* **440**

98. 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 access and write on the file
b. **pat** can rename the file
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

99. 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 < crontab.day**
c. **crontab -e crontab.day**
d. **crontab > crontab.day**
e. **echo crontab.day | crond**

100. 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. **pat** can access and write on the file
b. **bob** 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

101. 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. **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

102. What command changes a user's password?

a. **mkpasswd** *b.* **chpasswd** *c.* **passwd**
d. **chsh** *e.* **password**

103. 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 create a new file in the directory
b. **pat** can access and write on the file
c. **bob** can access and write on the file
d. **bob** can list names in the directory
e. **pat** can rename the file

104. Inside a shell script, which expands to the name of the script itself?

a. **"$?"** *b.* **"$#"** *c.* **"$@"** *d.* **"$0"** *e.* **"$*"**

105. What would be the output of the following command line:

```
echo a b c d | awk '{print $2}'
```

a. **a b** *b.* **b** *c.* no output
d. **c d** *e.* **$2**

106. Which of these safely tests for a null (empty) first argument?

a. **if [$1 = '']**
b. **if ["$1" = '']**
c. **if ["$1" -eq '/dev/null']**
d. **if [$1 = ""]**
e. **if ["$1" -eq '']**

107. Which command line below does not show any lines from inside the file **bat**?

a. **ls bat** *b.* **less bat** *c.* **tail bat**
d. **more bat** *e.* **head bat**

108. 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. **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 access and write on the file
e. **bob** can list names in the directory

109. Given my directory containing a file, which octal permissions allow me to delete the file from the directory, but not append data to the file?

a. Directory: `600` File: `300`
b. Directory: `700` File: `500`
c. Directory: `600` File: `500`
d. Directory: `500` File: `500`
e. Directory: `700` File: `200`

110. Which of the following options for `bash` or `sh` might be useful for debugging a shell script?

a. `-r` *b.* `-c` *c.* `-x` *d.* `-z` *e.* `-l`

111. 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 .`
c. `ls /tmp/dir/..`
d. `ls /tmp/dir/.`
e. `ls ..`

112. 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. `bob` can list names in the directory
b. `pat` can access and write on the file
c. `pat` can rename the file
d. `bob` can access and write on the file
e. `bob` can create a new file in the directory

113. What value `umask` gives a new file permissions `r--r-----`?

a. `440` *b.* `220` *c.* `447` *d.* `326` *e.* `110`

114. The `cron` system can run commands at most every

a. minute *b.* day *c.* hour
d. second *e.* millisecond

115. What value `umask` gives a new file permissions `r--r-----`?

a. `110` *b.* `446` *c.* `220` *d.* `226` *e.* `440`

116. 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. No, because the directory is not accessible to `bird`
b. No, because `foo` has no write permissions for `bird`
c. Yes; permissions don't apply because `bird` owns `foo`
d. No, because the directory has no write permissions for `bird`
e. Yes, because `bird` has read permissions on `foo`

117. 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/c`
c. `./1/2/a/b/c`
d. `./1/2/b/c`
e. `./1/2/3/a/b/c`

118. What would the following command do: `at 2pm`

a. read commands from stdin to be run every day at 2pm
b. read commands from stdin to be run once at 2pm
c. issue an error message
d. run the user's `crontab` jobs every day at 2pm
e. run the user's `crontab` jobs at 2pm

119. What command displays the kernel ring buffer of log messages:

a. `psmine` *b.* `ps lxww` *c.* `crontab`
d. `showall` *e.* `dmesg`

120. 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 /one/foo /two/bar`
b. `ln /two/bar /one/foo`
c. `ln /one/bar /two/foo`
d. `ln -s /two/bar /one/foo`
e. `ln -s /one/foo /two/bar`

121. 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. `rwx` on `a`, `wx` on `b`, none on `foo`
c. `wx` on `a`, `wx` on `b`, none on `foo`
d. `wx` on `a`, `wx` on `b`, `r` on `foo`
e. `rwx` on `a`, `wx` on `b`, `rw` on `foo`

122. Which command removes adjacent duplicate lines from a file?

a. `uniq` *b.* `dup` *c.* `dupl`
d. `unique` *e.* `duplicate`

123. Which command usually goes in your `.bash_profile` file?

a. `source ./.bashrc`
b. `./.bashrc source`
c. `cat ./.bashrc`
d. `source ./.bash_profile`
e. `./.bash_profile source`

124. If the line, `exit 2`
is executed in a shell script, what is the result?

a. an invalid argument error message
b. termination after sleeping for 2 seconds
c. the script breaks out of up to 2 levels of loops
d. termination with an exit status of 2
e. termination with an exit status of 0

125. Given my directory containing a file, which octal permissions allow me to delete the file from the directory, but not append data to the file?

a. Directory: `100` File: `200`
b. Directory: `300` File: `300`
c. Directory: `300` File: `500`
d. Directory: `100` File: `100`
e. Directory: `500` File: `400`

126. What permissions are given to `newfile` after this command line:

```
umask 326 ; touch newfile
```

a. `r--r-----` *b.* `r--r-x--x` *c.* `-wx-w-r-x`

d. `-wxr-----` *e.* `-wx-w-rw-`

127. 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. `bob` can list names in the directory

b. `pat` can rename the file

c. `bob` can create a new file in the directory

d. `pat` can create a new file in the directory

e. `bob` can access and write on the file

128. What is the output on your screen of the following sequence of commands:

```
i=00 ; [ $i -eq 0 ] ; echo $?
```

a. `test: $i: integer expression expected`

b. `1`

c. no output

d. `0`

e. the number 0 or 1 followed by another 0 or 1 on a new line

129. 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. Yes, because `bird`'s group matches the group writable directory

b. No, because `bird` cannot read the directory

c. No, because `bird` has no permissions on `foo`

d. No, because the directory has no permissions for other users

e. Yes; permissions don't apply because `bird` owns `foo`

130. Which test checks to see if the pathname is not an empty file (zero bytes)?

a. `test -e path` *b.* `test -s path` *c.* `test -x path`

d. `test -z path` *e.* `test -n path`

131. 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. `test: $a: string expression expected`

c. no output

d. `bash: 123: command not found`

e. `123`

132. What is the output on your screen of the following sequence of commands:

```
x=ok ; y=ok ; [ x = y ]
```

a. no output on screen

b. `1`

c. `test: x: integer expression expected`

d. `0`

e. `bash: x: command not found`

133. Which command line below does not show any lines from inside the file `out`?

a. `tail out` *b.* `sort out` *c.* `more out`

d. `wc out` *e.* `head out`

134. 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. 15 pathnames

b. an error message because `*/.` does not exist

c. 5 directory names

d. `*/.`

e. no output

135. The octal mode of a directory that allows the user to `cd` into it and list the names in it, but not to create any new files:

a. `100` *b.* `400` *c.* `200` *d.* `300` *e.* `500`

136. 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 list names in the directory

b. `pat` can access and write on the file

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

137. If `browser=lynx` then which one of the following `case` patterns will match this statement: `case "$browser" in`

a. `l?n? ) echo yes ;;`

b. `[lynx] | [LYNX] ) echo yes ;;`

c. `@ ) echo yes ;;`

d. `?lynx? ) echo yes ;;`

e. `(*ynx  echo yes ;;`

138. Which of the following would result in a "true" exit status?

a. `[ '00' -eq "0" ]` *b.* `[ 00 = 0 ]`

c. `[ '00' -ne "0" ]` *d.* `[ '00' = "0" ]`

e. `[ '00' != "00" ]`

139. Which expands to the exit status of the previous command?

a. `"$0"` *b.* `"$@"` *c.* `"$*"` *d.* `"$?"` *e.* `"$#"`

140. 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** matches the writable other permissions
b. Yes; permissions don't apply because **bird** owns **foo**
c. No, because the directory is not accessible to **bird**
d. Yes, because **bird** has full permissions on **foo**
e. No, because **bird** has no write permission on the directory

141. The **minimum** permissions you need to link a file **foo** from directory **a** to directory **b** are:

a. **rwx** on **a**, **wx** on **b**, **rw** on **foo**
b. **x** on **a**, **wx** on **b**, none on **foo**
c. **rwx** on **a**, **wx** on **b**, none on **foo**
d. **wx** on **a**, **wx** on **b**, **w** on **foo**
e. **wx** on **a**, **wx** on **b**, **r** on **foo**

142. 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 users in group **bar**
c. only user **foo**
d. only **root** can change the permissions
e. anyone except user **foo**

143. 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. **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

144. What command line shows only your own processes, not all processes?

a. **crontab** *b.* **psmine** *c.* **ps lxww**
d. **dmesg** *e.* **showall**

145. Inside a shell script, which expands to the number of script arguments?

a. **"$#"** *b.* **"$?"** *c.* **"$@"** *d.* **"$*"** *e.* **"$0"**

146. Given my directory containing a file, which octal permissions allow me to access and append data to the file but not delete the file?

a. Directory: **200** File: **200** *b.* Directory: **400** File: **400**
c. Directory: **100** File: **100** *d.* Directory: **600** File: **700**
e. Directory: **100** File: **200**

147. 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 access and write on the file
b. **bob** can list names in the directory
c. **pat** can rename the file
d. **pat** can access and write on the file
e. **bob** can create a new file in the directory

148. 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**

149. 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. **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

150. Which command displays all processes in a full wide listing?

a. **ps zxvf** *b.* **ps -all -wide**
c. **ps -any -wide** *d.* **ps laxww**
e. **ps -full**

151. What minimal permissions must you have on a directory to be able to execute successfully the command **ls .** from *inside* the directory?

a. **300** *b.* **400** *c.* **600** *d.* **100** *e.* **500**

152. 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. No, because the directory has no write permissions for others
b. No, because **foo** has no read permissions for **bird**
c. No, because the directory is not accessible to **bird**
d. Yes, because **bird** has write permissions on **foo**
e. Yes; permissions don't apply because **bird** owns **foo**

153. 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. **pat** can rename the file
c. **bob** can list names in the directory
d. **pat** can access and write on the file
e. **bob** can create a new file in the directory

154. The signal sent to a foreground process by typing the **[Ctrl-C]** key is:

a. **SIGINT** *b.* **SIGKILL** *c.* **SIGHUP**
d. **SIGSTOP** *e.* **SIGTERM**

155. 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**

156. 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. No, because **bird** has no write permission on the directory
b. No, because **bird** has no write permissions on **foo**
c. No, because the directory is not accessible to **bird**
d. No, because execute permissions are not set for **bird** on **foo**
e. Yes; permissions don't apply because **bird** owns **foo**

157. 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 execute permissions are not set for **bird** on **foo**
b. Yes; permissions don't apply because **bird** owns **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**

158. Under what directory are system configuration files usually stored?

a. **/var/log/** *b.* **/bin/** *c.* **/etc**
d. **/log/var/** *e.* **/usr/bin**

159. 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 *[Hh][Ii]*** *b.* **echo *(H,h,I,i)***
c. **echo ?[HhIiHhIi]?** *d.* **echo *[hiHI]***
e. **echo ?[HhIi]?**

160. 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 create a new file in the directory
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

161. The **minimum** permissions you need to copy a file **foo** from directory **a** to directory **b** are:

a. **wx** on **a**, **wx** on **b**, none on **foo**
b. **rwx** on **a**, **wx** on **b**, none on **foo**
c. **x** on **a**, **wx** on **b**, **r** on **foo**
d. **wx** on **a**, **wx** on **b**, **rw** on **foo**
e. **rx** on **a**, **wx** on **b**, **w** on **foo**

162. 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 create a new file in the directory
b. **bob** 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

163. 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 bar bar** *b.* **./foo "bar" 'bar'**
c. **./foo $3 $3** *d.* **./foo '$3' bar**
e. **./foo "$1" '$3'**

164. What command displays the groups you are in?

a. **grouprint** *b.* **groups** *c.* **mkgroups**
d. **gpasswd** *e.* **lstgroups**

165. 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**

166. The octal mode of a directory that allows the user to **cd** into it, but not to create any new files or to list any of the names in it:

a. **500** *b.* **100** *c.* **300** *d.* **400** *e.* **200**

167. Given the following shell script statement,

```
    if [ "a" = "b" ] ; then echo SAME ; fi
```

which of the following statements is true?

a. "**[**" is part of all "**if**" statements

b. "**[**" is passed four arguments

c. "**fi**" would cause a "command not found" error

d. an "invalid number" error would result

e. "**SAME**" would be printed

168. 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 list names in the directory

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 access and write on the file

169. Which command line would show the inode number of a file?

a. **ls -l file** *b.* **cat -l file** *c.* **find -i file**

d. **ls -i file** *e.* **cat -i file**

170. Given my directory containing a file, which octal permissions allow me to access and append data to the file but not delete the file?

a. Directory: **400** File: **400** *b.* Directory: **500** File: **600**

c. Directory: **100** File: **100** *d.* Directory: **300** File: **200**

e. Directory: **600** File: **700**

171. 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. Yes, because **bird** owns **foobar**

c. Yes, because **bird** has write permissions on **foobar**

d. No, because the directory is not accessible to **bird**

e. Yes, because **sesame** has write permissions on **foobar**

172. 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 222 dir**

b. **mkdir dir ; cd dir ; chmod ugo=w .**

c. **mkdir dir ; cd dir ; chmod ugo-rw .**

d. **mkdir dir ; chmod 333 dir**

e. **mkdir dir ; chmod 333 .**

173. What is the output of this command line in an empty directory:

```
    touch .a .b .c ; echo [.]*
```

a. no output

b. an error message from **echo** saying **[.]*** does not exist

c. **[.]***

d. **.a .b .c**

e. **. .. .a .b .c**

174. What would be the output of the following command line:

```
    echo a b c d | awk '{print $NF}'
```

a. **$NF** *b.* **d** *c.* no output

d. **a b c d** *e.* **4**

175. 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 current directory

e. a list of accounts in the password file

176. Given my directory containing a file, which octal permissions allow me to access and append data to the file but not delete the file?

a. Directory: **600** File: **700** *b.* Directory: **500** File: **100**

c. Directory: **100** File: **600** *d.* Directory: **400** File: **400**

e. Directory: **200** File: **200**

177. A Unix/Linux "tarball" is:

a. a single-file that contains individual compressed files

b. a multi-file directory containing individual compressed files

c. a single-file that contains individual uncompressed files

d. a multi-file directory containing individual uncompressed files

e. a single compressed file containing one uncompressed file

178. Which of these commands makes a file owned by me, also readable by me?

a. **chmod r=u ./myfile** *b.* **umask 300 ./myfile**

c. **chmod r+u myfile** *d.* **umask 400 myfile**

e. **chmod u+r ./myfile**

179. 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. **bob** can access and write on the file

c. **pat** can rename the file

d. **pat** can access and write on the file

e. **bob** can create a new file in the directory

180. 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 create a new file in the directory
e. **pat** can rename the file

181. 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. **./1/2/a/b** *b.* **./1/2/b/c** *c.* **./1/a/b**
d. **./a/b/c** *e.* **./1/2/c**

182. To show all your one-time scheduled commands, type:

a. **crontab -l** *b.* **/etc/crontab**
c. **/var/log/crontab** *d.* **atq**
e. **cat crontab**

183. 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. **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

184. The **minimum** permissions you need to delete a file **foo** from directory **a** are:

a. **wx** on **a**, **r** on **foo** *b.* **rwx** on **a**, **rw** on **foo**
c. **wx** on **a**, **w** on **foo** *d.* **rwx** on **a**, none on **foo**
e. **wx** on **a**, none on **foo**

185. 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 create a new file in the directory
b. **bob** can access and write on the file
c. **pat** can rename the file
d. **bob** can list names in the directory
e. **pat** can access and write on the file

186. 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. **ls: nosuchfile: No such file or directory**
c. **dog**
d. **bash: 2>/dev/null: command not found**
e. **nosuchfile**

187. What does the **-v** option to the **fgrep** command do?

a. prints the version number of the **fgrep** command
b. turns on the translation of unprintable characters
c. selects lines that do not contain a match for the supplied pattern
d. selects lines that do not contain unprintable characters
e. turns off the translation of unprintable characters

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

a. **132** *b.* **231** *c.* **123** *d.* **321** *e.* **312**