PRINT Name: ____________________________________

One-Answer Multiple Choice **155 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 ○Ⓑ○○○
192. Answer **192** is Ⓐ○○○○
193. Answer **193** is ○○○○Ⓔ
194. Answer **194** is ○○Ⓒ○○
195. Answer **195** is Ⓐ○○○○
196. Answer **196** is ○○○Ⓓ○

Your **Test Version** is:

B A E C 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. 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. **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 create a new file in the directory
e. **bob** can rename the file

2. 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 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

3. 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 **300** on directory **dir** and **400** on file **dir/bar**.
c. Permissions **100** on directory **dir** and **500** on file **dir/bar**.
d. Permissions **100** on directory **dir** and **300** on file **dir/bar**.
e. Permissions **300** on directory **dir** and **200** on file **dir/bar**.

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

5. 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 ..** *b.* **ls /tmp/dir/.**
c. **ls /tmp/dir** *d.* **ls .**
e. **ls /tmp/dir/..**

6. 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 -s /b/new /a/foo** *d.* **ln /b/new /a/foo**
e. **ln /a/new /b/foo**

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

8. 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 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

9. 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. **pat** can rename 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

10. 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 create a new file in the directory
b. **bob** can rename 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

11. 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`

12. Other than `root`, who can change the permissions of the following directory?

```
    dr-xrwxrwx 17 foo bar 4096 Apr 15 16:40 .
```

a. only users in group `bar`
b. only `root` can change the permissions
c. user `foo` and any user in group `bar`
d. only user `foo`
e. anyone except user `foo`

13. 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.* `"f"`
d. `c d e` *e.* `a b`

14. 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 `sesame` has no write permissions on `foobar`
b. Yes, because `bird` owns `foobar`
c. No, because the directory is not accessible to `bird`
d. Yes, because `bird` has write permissions on `foobar`
e. No, because execute permissions are not set for `bird` on `foobar`

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

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

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

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

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

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

19. The output of the `whoami` command is:

a. the current directory
b. your userid
c. a list of accounts in the password file
d. your HOME directory
e. a list of users logged in to the system

20. 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/b/bar`
d. `/tmp/a/foo` *e.* `/tmp/b/foo`

21. 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 `700` on directory `dir` and `200` on file `dir/bar`.
b. Permissions `600` on directory `dir` and `500` on file `dir/bar`.
c. Permissions `600` on directory `dir` and `300` on file `dir/bar`.
d. Permissions `300` on directory `dir` and `100` on file `dir/bar`.
e. Permissions `500` on directory `dir` and `500` on file `dir/bar`.

22. Dereference the following symlink `bar` into its equivalent absolute path:

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

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

23. 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. `bob` can list names in the directory
b. `pat` can access and write on the file
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

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

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

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

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

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

```
    ln -s ../b/../b/../../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**

28. 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 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

29. 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.* **3a** *c.* **2b** *d.* **3b** *e.* **2a**

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

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

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

33. 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. **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

34. 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**

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

36. The **minimum** permissions you need to move 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**, **rw** on **foo**
c. **wx** on **a**, **wx** on **b**, **r** on **foo**
d. **rwx** on **a**, **wx** on **b**, none on **foo**
e. **wx** on **a**, **wx** on **b**, **w** on **foo**

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

38. 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 333 .`
    c. `mkdir dir ; cd dir ; chmod ugo=w .`
    d. `mkdir dir ; chmod 222 dir`
    e. `mkdir dir ; cd dir ; chmod ugo-rw .`

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

40. 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 `100` on file `dir/bar`.
    d. Permissions `300` on directory `dir` and `500` on file `dir/bar`.
    e. Permissions `100` on directory `dir` and `200` on file `dir/bar`.

41. 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 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

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

43. When an `at` 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 current directory that was in use when the `at` job was created
    d. the directory with the name `/home`
    e. the system ROOT directory

44. 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 `300` on directory `dir` and `200` on file `dir/bar`.
    c. Permissions `500` on directory `dir` and `600` on file `dir/bar`.
    d. Permissions `600` on directory `dir` and `700` on file `dir/bar`.
    e. Permissions `100` on directory `dir` and `100` on file `dir/bar`.

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

46. 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 ; chmod 333 dir`
    c. `mkdir dir ; cd dir ; chmod go+wx .`
    d. `mkdir dir ; chmod 777 dir`
    e. `mkdir dir ; cd dir ; chmod go-x .`

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

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

49. 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 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

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

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

52. What command changes a user’s password?
a. `passwd` *b.* `mkpasswd` *c.* `chpasswd`
d. `chsh` *e.* `password`

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

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

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

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

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

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

57. 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. `./1/2/c` *b.* `./1/2/a/b/c`
c. `./1/a/b/c` *d.* `./1/2/b/c`
e. `./1/2/3/a/b/c`

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

59. 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 access and write on the file
b. `pat` 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 rename the file

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

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

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

63. 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 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

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

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

66. 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. `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 create a new file in the directory

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

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

68. 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 `200` on file `dir/bar`.
b. Permissions `600` on directory `dir` and `700` on file `dir/bar`.
c. Permissions `400` on directory `dir` and `400` on file `dir/bar`.
d. Permissions `200` on directory `dir` and `200` on file `dir/bar`.
e. Permissions `100` on directory `dir` and `100` on file `dir/bar`.

69. Can three different files have the same inode number on three different file systems?

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

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

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

a. `4` *b.* `$NF` *c.* `a b c d`
d. no output *e.* `d`

71. Process signals in increasing order of strength:

a. `KILL HUP TERM` *b.* `TERM KILL HUP`
c. `HUP KILL TERM` *d.* `TERM HUP KILL`
e. `HUP TERM KILL`

72. 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 `700` on directory `dir` and `200` on file `dir/bar`.
d. Permissions `600` on directory `dir` and `300` on file `dir/bar`.
e. Permissions `700` on directory `dir` and `500` on file `dir/bar`.

73. 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`

74. 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 access and write on the file
c. `bob` can rename the file
d. `pat` can access and write on the file
e. `bob` can list names in the directory

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

76. 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.* `root and bob` *c.* `bob`
d. `root` *e.* `foo`

77. 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. `ls: *: No such file or directory`
c. `dog`
d. `bash: 1>/dev/null: command not found`
e. no output

78. 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 `500` on directory `dir` and `100` on file `dir/bar`.
c. Permissions `200` on directory `dir` and `200` on file `dir/bar`.
d. Permissions `500` on directory `dir` and `200` on file `dir/bar`.
e. Permissions `600` on directory `dir` and `700` on file `dir/bar`.

79. 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 /home/foo/bar`
e. `ls .`

80. 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. `dog`
b. no output
c. `nosuchfile`
d. `ls: nosuchfile: No such file or directory`
e. `bash: 2>/dev/null: command not found`

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

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

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

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

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

84. Which **`crontab`** line executes at **`13:54`** every day?

a. `13 * * * 54 command` *b.* `* * * 54 13 command`
c. `13 54 * * * command` *d.* `54 13 * * * command`
e. `* * * 13 54 command`

85. 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 **`200`** on directory **`dir`** and **`200`** 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`**.

86. 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. **`pat`** can rename the file
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

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

a. `wc out` *b.* `sort out` *c.* `head out`
d. `tail out` *e.* `more out`

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

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

89. Which of these statements is true?

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

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

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

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

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

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

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

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

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

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

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

95. 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 -v 2` *e.* `at -l 2`

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

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

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

99. 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 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

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

101. 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 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 list names in the directory
e. `pat` can rename the file

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

103. 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. `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. `bob` can access and write on the file
e. `pat` can rename the file

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

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

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

107. 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. `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 list names in the directory

108. 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. `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. `bob` can create a new file in the directory
e. `pat` can rename the file

109. 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 > crontab.day`
b. `crontab -l crontab.day`
c. `echo crontab.day | crond`
d. `crontab -e crontab.day`
e. `crontab < crontab.day`

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

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

a. `chmod r+u myfile` *b.* `umask 300 ./myfile`
c. `umask 400 myfile` *d.* `chmod r=u ./myfile`
e. `chmod u+r ./myfile`

112. Which of the following signals is strongest (cannot be handled or ignored)?

a. `SIGINT` *b.* `SIGSUSP` *c.* `SIGTERM`
d. `SIGKILL` *e.* `SIGHUP`

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

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

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

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

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

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

117. What permissions are given to `newdir` after this command line:

```
umask 156 ; mkdir newdir
```

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

118. 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. 15 pathnames
c. no output
d. `*/.`
e. 5 directory names

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

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

120. 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. `pat` can access and write on the file
b. `pat` can rename the file
c. `bob` 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

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

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

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

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

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

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

125. 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 create a new file in the directory
b. **pat** can rename the file
c. **bob** 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

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

127. 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. **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

128. Which command usually goes in your **.bash_profile** file?

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

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

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

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

132. What command displays the groups you are in?

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

133. 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. **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

134. Which expands to all the script arguments?

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

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

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

136. Which line is from the Standard Script Header in this course?

a. `PATH=/bin:/usr/bin` *b.* `PATH=/bin:/urs/bin`
c. `PATH=/bin:/user/bin` *d.* `PATH=/bin:user/bin`
e. `PATH=/bin:ur/bin`

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

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

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

138. 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

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

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

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

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

142. 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. **pat** can access and write on the file
b. **pat** can rename the file
c. **bob** 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

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

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

a. `[.]*`
b. an error message from **echo** saying **[.]*** does not exist
c. no output
d. `.a .b .c`
e. `. .. .a .b .c`

144. 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.tar.gz`
b. `tar -tgz archive.tar.gz`
c. `tar -tgz archive`
d. `tar -tzf archive`
e. `tar -xzf archive.tar.gz`

145. 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 foo` *b.* `root and bob` *c.* `root`
d. `foo` *e.* `bob`

146. 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 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

147. 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 create a new file 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 list names in the directory

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

149. 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 the directory has no permissions for other users
b. No, because **bird** has no permissions on **foo**
c. No, because **bird** cannot read the directory
d. Yes, because **bird**'s group matches the group writable directory
e. Yes; permissions don't apply because **bird** owns **foo**

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

151. 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. Yes, because **bird** owns **foobar**
b. No, because the directory is not accessible to **bird**
c. Yes, because **sesame** has write permissions on **foobar**
d. Yes, because **bird** has write permissions on **foobar**
e. No, because execute permissions are not set for **bird** on **foobar**

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

153. 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 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

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

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