

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

Your Test Version is:

**E E C C A A**

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**.  

```
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. **bob** can access and write on the file
  - d. **bob** can list names in the directory
  - e. **pat** can access and write on the file
2. When a user named **bob** runs a command in an executable file owned by **foo**, in a directory owned by **root**, the file executes with the permissions of:
  - a. **bob**
  - b. **foo**
  - c. **root and bob**
  - d. **root**
  - e. **root and foo**
3. 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. 200
  - b. 100
  - c. 300
  - d. 400
  - e. 500
4. 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 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 rename the file
  - e. **pat** can access and write on the file

5. The **minimum** permissions you need to copy a file **foo** from directory **a** to directory **b** are:
  - a. **wx** on **a**, **wx** on **b**, **rw** on **foo**
  - b. **x** on **a**, **wx** on **b**, **r** on **foo**
  - c. **rx** on **a**, **wx** on **b**, **w** on **foo**
  - d. **rw**x on **a**, **wx** on **b**, none on **foo**
  - e. **wx** on **a**, **wx** on **b**, none on **foo**
6. Given the following shell script statement,

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

which of the following statements is true?
  - a. "**fi**" would cause a "command not found" error
  - b. an "invalid number" error would result
  - c. "[" is passed four arguments
  - d. "**SAME**" would be printed
  - e. "[" is part of all "**if**" statements
7. Inside a shell script, which expands to the name of the script itself?
  - a. "\$#"      b. "\$?"      c. "\$0"      d. "\$\*"      e. "\$@"
8. 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: 100 File: 600
  - c. Directory: 200 File: 200
  - d. Directory: 600 File: 700
  - e. Directory: 500 File: 100
9. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.  

```
dr-xr-xrwx 2 pat bg1 60 Jan 1 1:00 foo
-rwxrwxr-x 1 pat ted 0 Jan 1 1:00 foo/bar
```

  - a. **bob** can access and write on the file
  - b. **pat** can rename the file
  - c. **pat** can create a new file in the directory
  - d. **bob** can rename the file
  - e. **bob** can list names in the directory
10. Which of the following would result in a "true" exit status?
  - a. [ '00' -ne "0" ]
  - b. [ '00' -eq "0" ]
  - c. [ 00 = 0 ]
  - d. [ '00' != "00" ]
  - e. [ '00' = "0" ]
11. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.  

```
dr-xrwx-wx 2 pat pgg 60 Jan 1 1:00 foo
-r-xrwxr-x 1 bob bg2 0 Jan 1 1:00 foo/bar
```

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

12. 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: 600  
c. Directory: 100 File: 100                      d. Directory: 300 File: 200  
e. Directory: 400 File: 400
13. Under what directory are system configuration files usually stored?
- a. /log/var/                      b. /usr/bin                      c. /var/log/  
d. /etc                      e. /bin/
14. 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. 500                      c. 300                      d. 400                      e. 100
15. 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** has write permissions on **foobar**  
c. Yes, because **sesame** has write permissions on **foobar**  
d. No, because the directory is not accessible to **bird**  
e. Yes, because **bird** owns **foobar**
16. 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. 400                      b. 100                      c. 200                      d. 300                      e. 500
17. Which of the following commands would result in an error?
- a. [ 3 -eq 4 ]                      b. [ a != 4 ]                      c. [ 3 = 4 ]  
d. [ a -eq 4 ]                      e. [ a = 4 ]
18. 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 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
19. 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. 300                      c. 400                      d. 500                      e. 200

20. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
- ```
d--x-wx--- 2 bob pgg 60 Jan 1 1:00 foo
-r-x-w-r-x 1 bob bg1 0 Jan 1 1:00 foo/bar
```
- a. **bob** can 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
21. When an **at** job runs, the current working directory is set to:
- a. the HOME directory of the user who created the job  
b. the current directory that was in use when the **at** job was created  
c. the system ROOT directory  
d. the directory with the name **/root**  
e. the directory with the name **/home**
22. 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: 100 File: 500  
c. Directory: 300 File: 200                      d. Directory: 300 File: 400  
e. Directory: 100 File: 300
23. When a personal **crontab** job runs, the current working directory is set to:
- a. the system ROOT directory  
b. the directory with the name **/home**  
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
24. To bring a background shell job into the foreground, type:
- a. **bg**                      b. **fg**                      c. **kill %1**  
d. **[Ctrl-Z]**                      e. **[Ctrl-D]**
25. How does system logging work under Unix/Linux?
- a. processes write log files into each user's **\$HOME** directory  
b. processes send messages to a central **rsyslog** program that writes log files  
c. processes write log entries directly into the system log directory  
d. processes send messages to the **init** process that inherits orphan processes  
e. processes copy logs from your **\$HOME** directory to the **/var/spool** directory
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/a/b**                      b. **./a/b/c**                      c. **./1/2/b/c**  
d. **./1/2/c**                      e. **./1/2/a/b**

27. What is the output on your screen of the following sequence of commands:  
`x=pig ; [ -z $x ] ; echo $?`  
 a. the number 0 or 1 followed by another 0 or 1 on a new line  
 b. no output  
 c. 0  
 d. 1  
 e. `test: $x: integer expression expected`
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 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
29. 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. no output on screen  
 c. 0  
 d. 1  
 e. the number 1 or 0 followed by another 1 or 0 on a new line
30. 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. **pat** can access and write on the file  
 c. **bob** can list names in the directory  
 d. **pat** can rename the file  
 e. **bob** can access and write on the file
31. What minimal permissions must you have on a directory to be able to execute successfully the command `ls .` from *inside* the directory?  
 a. 500      b. 100      c. 600      d. 400      e. 300
32. 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 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

33. What command terminates processes based on their name (not safe!)?  
 a. `dmesg`      b. `kill`      c. `killall`  
 d. `ps lxxw`      e. `crontab`
34. 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. an invalid argument error message  
 c. termination after sleeping for 2 seconds  
 d. termination with an exit status of 2  
 e. termination with an exit status of 0
35. What command changes a user's password?  
 a. `password`      b. `chsh`      c. `chpasswd`  
 d. `passwd`      e. `mkpasswd`
36. Which of these outputs an error message on Standard Error?  
 a. `echo 1>$2 'error'`      b. `echo 2>$1 'error'`  
 c. `echo 1>&2 'error'`      d. `echo 1>2 'error'`  
 e. `echo 2>&1 'error'`
37. Which command line below does not show any lines from inside the file **bat**?  
 a. `ls bat`      b. `more bat`      c. `tail bat`  
 d. `head bat`      e. `less bat`
38. The **minimum** permissions you need to read a file **foo** in directory **a** are:  
 a. `rx` on **a**, `rw` on **foo**      b. `wx` on **a**, none on **foo**  
 c. `x` on **a**, `r` on **foo**      d. `rx` on **a**, none on **foo**  
 e. `wx` on **a**, `w` on **foo**
39. Dereference the following symlink **bar** into its equivalent absolute path:  
`ln -s ../b/../../a/./foo /tmp/a/b/bar`  
 a. `/tmp/a/foo`      b. `/tmp/foo`      c. `/tmp/b/bar`  
 d. `/tmp/a/b/bar`      e. `/tmp/b/foo`
40. The **minimum** permissions you need to link a file **foo** from directory **a** to directory **b** are:  
 a. `rx` on **a**, `wx` on **b**, `rw` on **foo**  
 b. `wx` on **a**, `wx` on **b**, `w` on **foo**  
 c. `wx` on **a**, `wx` on **b**, `r` on **foo**  
 d. `x` on **a**, `wx` on **b**, none on **foo**  
 e. `rx` on **a**, `wx` on **b**, none on **foo**
41. Which command displays all processes in a full wide listing?  
 a. `ps zxxvf`      b. `ps -all -wide`  
 c. `ps -any -wide`      d. `ps -full`  
 e. `ps laxww`

42. The signal sent to a foreground process by typing the **[Ctrl-C]** key is:  
 a. **SIGKILL**                      b. **SIGINT**                              c. **SIGTERM**  
 d. **SIGHUP**                        e. **SIGSTOP**
43. What would be the output of the following command line:  
`echo a b c d | awk '{print $2}'`  
 a. **\$2**                                b. **a b**                                      c. **c d**  
 d. **b**                                    e. no output
44. 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 -s /b/new /a/foo`  
 c. `ln /a/new /b/foo`                        d. `ln /b/new /a/foo`  
 e. `ln /a/foo /b/new`
45. What is the output on your screen of the following sequence of commands:  
`x=ok ; y=ok ; [ x = y ]`  
 a. **test: x: integer expression expected**  
 b. **1**  
 c. **0**  
 d. no output on screen  
 e. **bash: x: command not found**
46. 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 execute permissions are not set for **bird** on **foo**  
 c. Yes; permissions don't apply because **bird** owns **foo**  
 d. No, because the directory is not accessible to **bird**  
 e. No, because **bird** has no write permissions on **foo**
47. Given the following, can user **bird** in group **sesame** append to **./foo**?  
`dr-xr-xr-x 2 root sesame 4096 Oct 7 14:00 .`  
`-rw-r-xr-x 1 bird sesame 123 Oct 4 14:05 foo`  
 a. Yes; permissions don't apply because **bird** owns **foo**  
 b. Yes, because **bird** has write permissions on **foo**  
 c. No, because **bird** has no write permission on the directory  
 d. No, because execute permissions are not set for **bird** on **foo**  
 e. No, because the directory is not accessible to **bird**

48. 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 list names in the directory  
 b. **pat** can create a new file in the directory  
 c. **bob** can rename the file  
 d. **bob** can create a new file in the directory  
 e. **bob** can access and write on the file
49. Which of these statements is true?  
 a. You only need "**r--**" permission on directory "**foo**" for "**ls -l foo**" to work.  
 b. The "**ln**" command takes two arguments, so the maximum number of hard links a file can have is two.  
 c. To make a hard link to file "**foo**" named "**bar**", file "**foo**" must exist.  
 d. You can make a hard link to a directory.  
 e. If you give me write permission on a file owned by you, I can then use **chmod** to change its permissions.
50. 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 access and write on the file  
 c. **bob** can list names in the directory  
 d. **bob** can create a new file in the directory  
 e. **pat** can access and write on the file
51. 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 day  
 c. at 6:00am every day  
 d. at 12:06am every business day  
 e. at 6:00am every business day
52. 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. **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

53. 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`
- pat** can create a new file in the directory
  - pat** can rename the file
  - bob** can list names in the directory
  - bob** can rename the file
  - bob** can access and write on the file
54. What command line shows only your own processes, not all processes?
- crontab**
  - dmesg**
  - ps lxww**
  - showall**
  - psmine**
55. 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:
- 400**
  - 200**
  - 100**
  - 300**
  - 500**
56. In a shell **case** structure, the **case** segment that will GLOB match the text **a**, **b**, or **c**, is coded as
- a,b,c )**
  - a\b\c )**
  - a|b|c )**
  - a:b:c )**
  - a/b/c )**
57. Which command counts the number of Unix permission groups you are in?
- groups | wc**
  - echo groups | wc**
  - id | wc**
  - umask | wc**
  - wc groups**
58. Which of the following could you use as options for the **tar** command to extract a **gzip**-compressed archive?
- ezf**
  - xzf**
  - tgz**
  - czf**
  - egf**
59. What command would you use to see the command that **at** job number **2** will run?
- at -l 2**
  - at -m 2**
  - at -c 2**
  - at -v 2**
  - atq 2**
60. The *difference* between the system (**root**) crontab and all the user (personal) crontabs is:
- the personal crontab has the date and time in it
  - the personal crontab also has the userid in it
  - the system crontab has the date and time in it
  - the personal crontab only runs commands once
  - the system crontab also has the userid in it
61. 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?
- `./foo "bar" 'bar'`
  - `./foo "$1" '$3'`
  - `./foo '$3' bar`
  - `./foo bar bar`
  - `./foo $3 $3`

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)?
- `echo *[hiHI]*`
  - `echo *[Hh][Ii]*`
  - `echo ?[HhIiHhIi]?`
  - `echo ?[HhIi]?`
  - `echo *(H,h,I,i)*`
63. 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
```
- No, because **bird** has no permissions on **foo**
  - No, because **bird** cannot read the directory
  - No, because the directory has no permissions for other users
  - Yes, because **bird**'s group matches the group writable directory
  - Yes; permissions don't apply because **bird** owns **foo**
64. The **minimum** permissions you need to move a file **foo** from directory **a** to directory **b** are:
- wx** on **a**, **wx** on **b**, none on **foo**
  - wx** on **a**, **wx** on **b**, **r** on **foo**
  - rw**x on **a**, **wx** on **b**, **rw** on **foo**
  - wx** on **a**, **wx** on **b**, **w** on **foo**
  - rw**x on **a**, **wx** on **b**, none on **foo**
65. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.  
`d--xr----x 2 bob ted 60 Jan 1 1:00 foo`  
`-r-x-w-rwx 1 pat bg2 0 Jan 1 1:00 foo/bar`
- bob** can access and write on the file
  - pat** can rename the file
  - bob** can list names in the directory
  - bob** can create a new file in the directory
  - pat** can access and write on the file
66. Which expands to the exit status of the previous command?
- "\$0"**
  - "\$?"**
  - "\$\*"**
  - "\$#"**
  - "\$@"**
67. Which line is from the Standard Script Header in this course?
- PATH=/bin:/usr/bin**
  - PATH=/bin:/urs/bin**
  - PATH=/bin:user/bin**
  - PATH=/bin:/user/bin**
  - PATH=/bin:ur/bin**
68. In an empty directory, what permissions are on file **???** after these commands:  
`touch ??? *** ; chmod 111 *`  
`chmod 222 ??? ; chmod 444 '***'`
- `--x--x--x`
  - `-wx-wx-wx`
  - `rw-rw-rw-`
  - `r--r--r--`
  - `-w--w--w-`

69. Can three different files have the same inode number on three different file systems?
- no: inode numbers are unique across all file systems
  - yes: if the files are all names for the same inode
  - no: you can't have inode numbers on three file systems
  - no: inode numbers only apply to directories, not files
  - yes: inode numbers are only unique inside a file system
70. 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?
- `crontab < crontab.day`
  - `echo crontab.day | crond`
  - `crontab -l crontab.day`
  - `crontab > crontab.day`
  - `crontab -e crontab.day`
71. In a directory containing one file named `dog`, what is the output on your screen after this command line: `2>/dev/null ls nosuchfile`
- `ls: nosuchfile: No such file or directory`
  - `nosuchfile`
  - `bash: 2>/dev/null: command not found`
  - `dog`
  - no output
72. Which of the following options for `bash` or `sh` might be useful for debugging a shell script?
- `-c`
  - `-r`
  - `-z`
  - `-l`
  - `-x`
73. What value `umask` gives a new file permissions `r--r-----`?
- `446`
  - `110`
  - `220`
  - `440`
  - `226`
74. What minimal permissions must you have on a directory to be able to execute successfully the command `ls .` from *inside* the directory?
- `rw-`
  - `r--`
  - `--x`
  - `-wx`
  - `r-x`
75. Which of these commands makes a file owned by me, also readable by me?
- `umask 400 myfile`
  - `chmod r+u myfile`
  - `chmod u+r ./myfile`
  - `chmod r=u ./myfile`
  - `umask 300 ./myfile`
76. What value `umask` gives a new file permissions `r--r-----`?
- `440`
  - `326`
  - `447`
  - `220`
  - `110`
77. 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?
- `2a`
  - `2b`
  - `3a`
  - `3b`
  - `4c`

78. 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`
- `./1/2/c`
  - `./1/2/3/a/b/c`
  - `./1/a/b/c`
  - `./1/2/b/c`
  - `./1/2/a/b/c`
79. 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`
- `pat` can rename the file
  - `bob` can access and write on the file
  - `bob` can create a new file in the directory
  - `bob` can list names in the directory
  - `pat` can access and write on the file
80. 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`
- `pat` can rename the file
  - `bob` can create a new file in the directory
  - `bob` can list names in the directory
  - `bob` can access and write on the file
  - `pat` can create a new file in the directory
81. To list your personal crontab, type:
- `cat crontab`
  - `/etc/crontab`
  - `atq`
  - `/var/log/crontab`
  - `crontab -l`
82. Dereference the following symlink `bar` into its equivalent absolute path:  
`ln -s ../b/../../b/../../foo /tmp/a/b/bar`
- `/tmp/b/foo`
  - `/tmp/b/bar`
  - `/tmp/a/b/bar`
  - `/tmp/foo`
  - `/tmp/a/foo`
83. What value to `chmod` would change the permissions on a file to `r-----rw-`?
- `406`
  - `322`
  - `122`
  - `102`
  - `654`
84. Other than `root`, who can change the permissions of the following directory?  
`dr-xrwxrwx 17 foo bar 4096 Apr 15 16:40 .`
- only users in group `bar`
  - user `foo` and any user in group `bar`
  - anyone except user `foo`
  - only `root` can change the permissions
  - only user `foo`

85. 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`
- pat** can access and write on the file
  - pat** can rename the file
  - bob** can create a new file in the directory
  - bob** can access and write on the file
  - bob** can list names in the directory
86. Which command sequence correctly searches for the **string** and then prints **OK** if it is found inside the password file?
- `if [ test string /etc/passwd ] ; then echo OK ; fi`
  - `if test string /etc/passwd ; then echo OK ; fi`
  - `if fgrep string /etc/passwd ; then echo OK ; fi`
  - `if [ fgrep string /etc/passwd ] ; then echo OK ; fi`
  - `if test string = /etc/passwd ; then echo OK ; fi`
87. 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`
- pat** can access and write on the file
  - bob** can create a new file in the directory
  - bob** can list names in the directory
  - pat** can rename the file
  - bob** can access and write on the file
88. 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?
- 1 iteration
  - 7 iterations
  - 9 iterations
  - 6 iterations
  - 8 iterations
89. 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?
- `ls /tmp/dir/..`
  - `ls ..`
  - `ls .`
  - `ls /tmp/dir`
  - `ls /tmp/dir/.`
90. What does the **-v** option to the **fgrep** command do?
- turns off the translation of unprintable characters
  - prints the version number of the **fgrep** command
  - turns on the translation of unprintable characters
  - selects lines that do not contain a match for the supplied pattern
  - selects lines that do not contain unprintable characters

91. 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`
- bob** can rename the file
  - bob** can access and write on the file
  - bob** can list names in the directory
  - bob** can create a new file in the directory
  - pat** can access and write on the file
92. Given my directory containing a file, which octal permissions allow me to access and append data to the file but not delete the file?
- Directory: 400 File: 400
  - Directory: 100 File: 100
  - Directory: 100 File: 200
  - Directory: 600 File: 700
  - Directory: 200 File: 200
93. 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?
- 2a
  - 3b
  - 2b
  - 3a
  - 4c
94. 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?
- c d e
  - "f"
  - \$3
  - a b
  - f
95. 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`
- ./1/2/3/a/b
  - ./1/2/3/c
  - ./1/2/a/b
  - ./1/2/3/b/c
  - ./a/b/c
96. Inside a shell script, which correctly expands to be the first script argument without processing any special characters in the argument?
- \$1
  - "\$1"
  - '\$1'
  - \\$1
  - "\$1"
97. What value **umask** gives a new file permissions **r--r-----**?
- 446
  - 440
  - 110
  - 220
  - 337
98. What would be the output of the following command line:  
`echo a b c d | awk '{print $NF}'`
- a b c d
  - no output
  - d
  - \$NF
  - 4

99. What permissions are given to **newdir** after this command line:  
`umask 156 ; mkdir newdir`  
 a. `rw--w---x`                      b. `--xr-xrw-`                      c. `r-x-w-rw-`  
 d. `r-x--x---`                      e. `rw--w----`
100. 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. **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 access and write on the file  
 e. **bob** can access and write on the file
101. What is the output of this command line in an empty directory:  
`touch .a .b .c ; echo [.]*`  
 a. an error message from **echo** saying `[.]*` does not exist  
 b. `.a .b .c`  
 c. `. . . .a .b .c`  
 d. `[.]*`  
 e. no output
102. Given the following, can user **bird** in group **sesame** append to **foobar**?  
`drwx--xrw 2 root sesame 4096 Oct 7 14:00 .`  
`-rw----- 1 bird sesame 1024 Oct 4 14:05 foobar`  
 a. Yes, because **bird** owns **foobar**  
 b. Yes, because **bird** has write permissions on **foobar**  
 c. No, because execute permissions are not set for **bird** on **foobar**  
 d. No, because the directory is not accessible to **bird**  
 e. No, because **sesame** has no write permissions on **foobar**
103. Which of the following signals is strongest (cannot be handled or ignored)?  
 a. **SIGTERM**                      b. **SIGSUSP**                      c. **SIGINT**  
 d. **SIGKILL**                      e. **SIGHUP**
104. Given the following, can user **bird** in group **sesame** append to `./foo`?  
`dr-xr--r-x 2 root sesame 4096 Oct 7 14:00 .`  
`-rw-rw-r-- 1 bird sesame 123 Oct 4 14:05 foo`  
 a. No, because the directory is not accessible to **bird**  
 b. No, because **bird** has no write permission on the directory  
 c. Yes; permissions don't apply because **bird** owns **foo**  
 d. Yes, because **bird** has write permissions on **foo**  
 e. No, because execute permissions are not set for **bird** on **foo**

105. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.  
`dr-xrwx--x 2 pat pgg 60 Jan 1 1:00 foo`  
`--w----r-x 1 bob bg2 0 Jan 1 1:00 foo/bar`  
 a. **pat** can rename the file  
 b. **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. **bob** can create a new file in the directory
106. Which test checks to see if the pathname is not an empty file (zero bytes)?  
 a. `test -x path`                      b. `test -z path`                      c. `test -e path`  
 d. `test -n path`                      e. `test -s path`
107. 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 **foo** has no read permissions for **bird**  
 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 write permissions on **foo**  
 e. No, because the directory has no write permissions for others
108. The **minimum** permissions you need to delete a file **foo** from directory **a** are:  
 a. **rw**x on **a**, none on **foo**                      b. **w**x on **a**, **r** on **foo**  
 c. **w**x on **a**, **w** on **foo**                      d. **w**x on **a**, none on **foo**  
 e. **rw**x on **a**, **rw** on **foo**
109. The **minimum** permissions you need to append to a file **foo** in directory **a** are:  
 a. **rw**x on **a**, **rw** on **foo**                      b. **rw**x on **a**, none on **foo**  
 c. **w**x on **a**, **w** on **foo**                      d. **w**x on **a**, none on **foo**  
 e. **x** on **a**, **w** on **foo**
110. 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 list names in the directory  
 b. **pat** 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 rename the file
111. 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. `*/.`  
 b. an error message because `*/.` does not exist  
 c. 15 pathnames  
 d. no output  
 e. 5 directory names



112. 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:
- root** and **foo**
  - foo**
  - bob**
  - root**
  - root** and **bob**
113. 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`
- pat** can access and write on the file
  - bob** can create a new file in the directory
  - bob** can access and write on the file
  - bob** can list names in the directory
  - pat** can rename the file
114. If **guru=linus** then which one of the following **case** patterns will match this statement: **case "\$guru" in**
- [linus] | [LINUS] ) echo yes ; ;**
  - (\*nus echo yes ; ;**
  - lin? ) echo yes ; ;**
  - "linu?" ) echo yes ; ;**
  - \* ) echo yes ; ;**
115. What is the output on your screen of the following command sequence:  
`i=04; test $i = 4 ; echo $?`
- test: \$i: integer expression expected**
  - 0**
  - 1**
  - the number 0 or 1 followed by another 0 or 1 on a new line
  - no output
116. What command displays the groups you are in?
- mkgroups**
  - groups**
  - groupprint**
  - gpaswd**
  - lstgroups**
117. A Unix/Linux "tarball" is:
- a single compressed file containing one uncompressed file
  - a single-file that contains individual uncompressed files
  - a multi-file directory containing individual uncompressed files
  - a multi-file directory containing individual compressed files
  - a single-file that contains individual compressed files
118. In an empty directory, what permissions are on file **???** after these commands:  
`touch ??? *** ; chmod 111 *`  
`chmod 222 ? ; chmod 444 '*'`
- rw-rw-rw-**
  - wx-wx-wx**
  - x--x--x**
  - r--r--r--**
  - w--w--w-**

119. 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`
- pat** can rename the file
  - pat** can access and write on the file
  - bob** can access and write on the file
  - bob** can create a new file in the directory
  - bob** can list names in the directory
120. To show all your one-time scheduled commands, type:
- /etc/crontab**
  - /var/log/crontab**
  - cat crontab**
  - crontab -l**
  - atq**
121. 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`
- bob** can list names in the directory
  - bob** can create a new file in the directory
  - bob** can access and write on the file
  - pat** can rename the file
  - pat** can access and write on the file
122. Process signals in increasing order of strength:
- HUP KILL TERM**
  - TERM HUP KILL**
  - TERM KILL HUP**
  - KILL HUP TERM**
  - HUP TERM KILL**
123. Given my directory containing a file, which octal permissions allow me to access and append data to the file but not delete the file?
- Directory: **600** File: **700**
  - Directory: **400** File: **400**
  - Directory: **500** File: **200**
  - Directory: **200** File: **200**
  - Directory: **500** File: **100**
124. 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`
- bob** can create a new file in the directory
  - pat** can access and write on the file
  - bob** can access and write on the file
  - bob** can list names in the directory
  - pat** can rename the file

125. 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. 1 2 3 4                      b. 1234                      c. 1  
d. 10                              e. 4321
126. Which command line would show the inode number of a file?
- a. **find -i file**              b. **cat -i file**              c. **ls -i file**  
d. **cat -l file**                e. **ls -l file**
127. 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 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
128. 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 list names in the directory  
b. **pat** can rename the file  
c. **bob** can access and write on the file  
d. **bob** can rename the file  
e. **bob** can create a new file in the directory
129. 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 -s /one/foo /two/bar**  
c. **ln /two/bar /one/foo**  
d. **ln -s /two/bar /one/foo**  
e. **ln /one/foo /two/bar**
130. 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 = /dev/null ] ; then date ; fi**  
b. **if test "" -eq \$a ; then date ; fi**  
c. **if [ '' = \$a ] ; then date ; fi**  
d. **if test "" = "\$a" ; then date ; fi**  
e. **if [ "\$a" = \* ] ; then date ; fi**

131. 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. **pat** can rename the file  
b. **bob** can access and write on the file  
c. **pat** can access and write on the file  
d. **bob** can list names in the directory  
e. **bob** can create a new file in the directory
132. What value **umask** gives a new directory permissions **rw--w---x**?
- a. **621**                      b. **211**                      c. **156**                      d. **421**                      e. **432**
133. 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: 100                      b. Directory: 300 File: 300  
c. Directory: 300 File: 500                      d. Directory: 100 File: 200  
e. Directory: 500 File: 400
134. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.
- ```
d-w---xr-- 2 pat ted 60 Jan 1 1:00 foo
-rwxrwxrwx 1 pat bg2 0 Jan 1 1:00 foo/bar
```
- a. **bob** can rename the file  
b. **bob** can access and write on the file  
c. **bob** can create a new file in the directory  
d. **bob** can list names in the directory  
e. **pat** can access and write on the file
135. 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 has no write permissions for **bird**  
b. Yes; permissions don't apply because **bird** owns **foo**  
c. No, because the directory is not accessible to **bird**  
d. No, because **foo** has no write permissions for **bird**  
e. Yes, because **bird** has read permissions on **foo**
136. The output of the **whoami** command is:
- a. a list of accounts in the password file  
b. the current directory  
c. your HOME directory  
d. your userid  
e. a list of users logged in to the system

137. If **browser=lynx** then which one of the following **case** patterns will match this statement: **case "\$browser" in**
- [lynx] | [LYNX] ) echo yes ;;**
  - l?n? ) echo yes ;;**
  - @ ) echo yes ;;**
  - (\*ynx echo yes ;;**
  - ?lynx? ) echo yes ;;**
138. If **archive.tar.gz** is a compressed tar archive, which command could you run to produce a listing of its contents without extracting it?
- tar -tgz archive.tar.gz**
  - tar -tzf archive**
  - tar -xzf archive.tar.gz**
  - tar -tzf archive.tar.gz**
  - tar -tgz archive**
139. If **a=123** and **b=456** then what is the output of the following sequence of commands: **if [\$a = \$b] ; then echo \$a ; fi**
- bash: [123: command not found**
  - test: a=123: integer expression expected**
  - 123**
  - no output**
  - test: \$a: string expression expected**
140. 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**
- pat** can create a new file in the directory
  - bob** can list names in the directory
  - bob** can access and write on the file
  - pat** can rename the file
  - bob** can rename the file
141. If **a=123** and **b=456** then what is the output of the following sequence of commands: **if [ \$a = \$b ]; then echo \$a ; fi**
- 123**
  - test: a=123: integer expression expected**
  - test: \$a: string expression expected**
  - bash: 123: command not found**
  - no output**

142. 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**
- bob** can access and write on the file
  - bob** can list names in the directory
  - bob** can create a new file in the directory
  - pat** can rename the file
  - pat** can access and write on the file
143. To change your own account password, type this into the shell prompt:
- \$ passwd idallen-ubuntu**
  - \$ passwd**
  - \$ passwd .**
  - \$ passwd cst8207**
  - \$ passwd \***
144. 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**
- bob** can access and write on the file
  - bob** can rename the file
  - pat** can rename the file
  - bob** can list names in the directory
  - pat** can create a new file in the directory
145. What permissions are given to **newfile** after this command line:  
**umask 326 ; touch newfile**
- wxr-----**
  - r--r-----**
  - wx-w-r-x**
  - wx-w-rw-**
  - r--r-x--x**
146. What would the following command do: **at 2pm**
- read commands from stdin to be run once at 2pm
  - issue an error message
  - run the user's **crontab** jobs at 2pm
  - read commands from stdin to be run every day at 2pm
  - run the user's **crontab** jobs every day at 2pm
147. What value to **chmod** would change the permissions on a file to **rw-r--r--?**
- 644**
  - 244**
  - 311**
  - 344**
  - 211**
148. Which of the following commands would result in an error?
- [ a != 4 ]**
  - [ 3 -eq 4 ]**
  - [ 3 = f ]**
  - [ a = 4 ]**
  - [ 3 -e 3 ]**

149. 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`
- bob** can access and write on the file
  - bob** can create a new file in the directory
  - pat** can rename the file
  - pat** can access and write on the file
  - bob** can list names in the directory
150. 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?
- `mkdir dir ; chmod 333 dir`
  - `mkdir dir ; chmod 333 .`
  - `mkdir dir ; cd dir ; chmod ugo-rw .`
  - `mkdir dir ; cd dir ; chmod ugo=w .`
  - `mkdir dir ; chmod 222 dir`
151. Dereference the following symlink **bar** into its equivalent absolute path:  
`ln -s ../b/../../a/./foo /tmp/a/b/bar`
- `/tmp/a/foo`
  - `/tmp/b/foo`
  - `/tmp/a/b/bar`
  - `/tmp/foo`
  - `/tmp/b/bar`
152. Which **crontab** line executes at 13:54 every day?
- `* * * 54 13 command`
  - `13 54 * * * command`
  - `13 * * * 54 command`
  - `54 13 * * * command`
  - `* * * 13 54 command`
153. Which command line below does not show any lines from inside the file **out**?
- `tail out`
  - `more out`
  - `wc out`
  - `sort out`
  - `head out`
154. 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`
- bob** can access and write on the file
  - pat** can rename the file
  - pat** can access and write on the file
  - bob** can create a new file in the directory
  - bob** can list names in the directory
155. Which command sequence correctly compares the two numbers and prints **OK**?
- `if ( 3 < 4 ) ; then echo OK ; fi`
  - `if [ ! 4 -gt 3 ] ; then echo OK ; fi`
  - `if [ 4 > 3 ] ; then echo OK ; fi`
  - `if [ 4 -ge 3 ] ; then echo OK ; fi`
  - `if ( ! 4 < 3 ) ; then echo OK ; fi`
156. What value **umask** gives a new file permissions **r--r-----**?
- 110
  - 220
  - 440
  - 446
  - 237

157. Which expands to all the script arguments?
- `"$*"`
  - `"$0"`
  - `"$!"`
  - `"$#"`
  - `"$?"`
158. If **a=123** and **b=456** then what is the output of the following sequence of commands: `if $a = $b ; then echo $a ; fi`
- `bash: 123: command not found`
  - `test: a=123: integer expression expected`
  - no output
  - `test: $a: string expression expected`
  - 123
159. 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?
- `mkdir dir ; cd dir ; chmod go+wx .`
  - `mkdir dir ; chmod 777 .`
  - `mkdir dir ; cd dir ; chmod go-x .`
  - `mkdir dir ; chmod 333 dir`
  - `mkdir dir ; chmod 777 dir`
160. 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:
- 200
  - 500
  - 100
  - 400
  - 300
161. 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`
- No, because **bird** has no write permission on the directory
  - Yes, because **bird** has full permissions on **foo**
  - No, because the directory is not accessible to **bird**
  - Yes, because **bird** matches the writable other permissions
  - Yes; permissions don't apply because **bird** owns **foo**
162. What command manipulates your personal list of repeated scheduled commands:
- `psmine`
  - `showall`
  - `crontab`
  - `ps lxwx`
  - `dmesg`
163. What is the output on your screen of the following sequence of commands:  
`i=00 ; [ $i -eq 0 ] ; echo $?`
- no output
  - 1
  - the number 0 or 1 followed by another 0 or 1 on a new line
  - 0
  - `test: $i: integer expression expected`
164. 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`
- `./1/2/3/a/b/c`
  - `./1/2/3/a/b`
  - `./1/2/a`
  - `./1/a`
  - `./1/2/a/b/c`

165. 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`
- bob** can rename the file
  - bob** can access and write on the file
  - pat** can rename the file
  - pat** can create a new file in the directory
  - bob** can list names in the directory
166. 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`
- bob** can access and write on the file
  - pat** can create a new file in the directory
  - bob** can list names in the directory
  - bob** can rename the file
  - pat** can rename the file
167. 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`
- bob** can access and write on the file
  - bob** can list names in the directory
  - pat** can access and write on the file
  - bob** can create a new file in the directory
  - pat** can rename the file
168. 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?
- Directory: 500 File: 500
  - Directory: 700 File: 200
  - Directory: 300 File: 100
  - Directory: 600 File: 300
  - Directory: 600 File: 500
169. What command displays the kernel ring buffer of log messages:
- crontab**
  - dmesg**
  - ps lxww**
  - psmine**
  - showall**
170. Which command removes adjacent duplicate lines from a file?
- dupl**
  - duplicate**
  - unique**
  - dup**
  - uniq**
171. 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:
- 300
  - 500
  - 100
  - 400
  - 200
172. Under what directory are system log files usually stored?
- /log/var**
  - /var/log**
  - /bin/**
  - /usr/bin**
  - /etc/log**

173. What is the output on your screen of the following sequence of commands:  
`a=4 ; b=4 ; [ $a -le $b ] ; echo $?`
- 1
  - the number 1 or 0 followed by another 1 or 0 on a new line
  - test: \$a: integer expression expected**
  - 0
  - no output
174. Which of these safely tests for a null (empty) first argument?
- if [ \$1 = ' ]**
  - if [ "\$1" = ' ]**
  - if [ "\$1" -eq ' ]**
  - if [ "\$1" -eq '/dev/null' ]**
  - if [ \$1 = "" ]**
175. 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?
- Directory: 600 File: 500
  - Directory: 500 File: 500
  - Directory: 700 File: 200
  - Directory: 600 File: 300
  - Directory: 700 File: 500
176. Which command usually goes in your **.bash\_profile** file?
- source ./bashrc**
  - ./bash\_profile source**
  - source ./bash\_profile**
  - cat ./bashrc**
  - ./bashrc source**
177. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.  
`d--xrx-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`
- pat** can rename the file
  - bob** can access and write on the file
  - bob** can list names in the directory
  - bob** can create a new file in the directory
  - pat** can access and write on the file
178. To send a **KILL** signal to a process with process ID **PID**, which of the following commands would you use?
- kill -KILL PID**
  - send -KILL PID**
  - signal -KILL PID**
  - kill PID KILL**
  - send PID KILL**

179. User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.  
`d--xrw--x 2 bob ted 60 Jan 1 1:00 foo`  
`----rw--w- 1 bob bg1 0 Jan 1 1:00 foo/bar`
- bob** can access and write on the file
  - bob** can list names in the directory
  - pat** can access and write on the file
  - bob** can create a new file in the directory
  - pat** can rename the file
180. In a directory containing one file named **dog**, what is the output on your screen after this command line: `1>/dev/null ls *`
- \*
  - no output
  - `ls: *: No such file or directory`
  - `bash: 1>/dev/null: command not found`
  - dog**
181. The **cron** system can run commands at most every
- second
  - millisecond
  - hour
  - day
  - minute
182. Which of these statements is true?
- you can change the permissions of any file to which you can write
  - you can only make links to files owned by you
  - you may be able to rename a file even if you do not own the file
  - you can only remove a file name if the file is owned by you
  - you can only remove a file name if the file is writable by you
183. 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?
- `ls ..`
  - `ls /home/foo/bar`
  - `ls /home/foo/bar/..`
  - `ls .`
  - `ls /home/foo/bar/.`
184. 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"`
- the 'pig' ate
  - the '\$animal' ate
  - the 'animal' ate
  - the \$animal ate
  - the 'dog' ate
185. 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.
- `#!/bin/sh -u`
  - `#!/bin/sh -u`
  - `#!/bin/sh -u`
  - `!/bin/sh -u`
  - `!!/bin/sh -u`

186. Inside a shell script, which expands to the number of script arguments?
- `"$0"`
  - `"$@"`
  - `"$*"`
  - `"$#"`
  - `"$?"`
187. 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`
- No, because **foo** has no read permissions for **bird**
  - No, because the directory has no write permissions for **bird**
  - Yes, because **bird** has write permissions on **foo**
  - Yes; permissions don't apply because **bird** owns **foo**
  - No, because the directory is not readable by **bird**
188. **Did you read all the words of the test instructions on page one?**
- 231
  - 312
  - 321
  - 132
  - 123