

PRINT Name: \_\_\_\_\_

**One-Answer Multiple Choice 180 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 full Name 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 these test instructions is **jes** spelled backwards.

1. [13/75] If `d='Assignments/assignment11'` then which of these symbolic links enables the script `partA.sh` to run as my own command named `partA`?
  - a. `ln -s ../../CST8207-19W/$d/partA.sh ~/bin/partA`
  - b. `ln -s ./CST8207-19W/$d/partA.sh ~/bin/partA`
  - c. `ln -s CST8207-19W/$d/partA.sh ~/bin/partA`
  - d. `ln -s ../CST8207-19W/./$d/partA.sh ~/bin/partA`
  - e. `ln -s ../CST8207-19W/$d/partA.sh ~/bin/partA`
2. [16/76] In an empty directory, what is the output on your screen of this unquoted set of three commands:
 

```
mkdir a b c ; touch a/34 a/56 a/?? 345 456 56
find a -name ??
```

  - a. `a/56`
  - b. `a/??`
  - c. `a/34`
  - d. no output
  - e. `a/?? a/34 a/56`
3. [24/76] Change an existing directory `dir` into one where anyone can put a file, but nobody can see the names of the files that are there:
  - a. `chmod ugo+x dir ; chmod 222 dir`
  - b. `chmod 333 dir ; chmod ugo-x dir`
  - c. `chmod 012 dir ; chmod ugo+wx dir`
  - d. `chmod 222 dir`
  - e. `cd dir ; chmod 333 dir`
4. [27/76] What is the correct syntax to redirect both standard output and standard error into the same output file?
  - a. `cmd 1>out 2>out`
  - b. `cmd 2>&1 >out`
  - c. `cmd 2>1 >out`
  - d. `cmd 1>out 2>1`
  - e. `cmd >out 2>&1`

5. [27/76] What is the output of this in an empty directory: `ls *`
  - a. an error message from `ls` saying `*` does not exist
  - b. `. .`
  - c. an error message from `bash` saying `*` does not exist
  - d. no output on screen
  - e. `*`
6. [29/76] What permissions are given to `newfile` after this:
 

```
umask 362 ; touch newfile
```

  - a. `-wx---r--`
  - b. `r----xr-x`
  - c. `-wxrw--w-`
  - d. `-wxr-x-w-`
  - e. `r-----r--`
7. [31/76] In an empty directory, how many words are in file `e` after this:
 

```
echo Isn't >a Pat's. It's >c Lee's. >d ; ls >e
```

  - a. 2
  - b. 1
  - c. 3
  - d. 0
  - e. 4
8. [33/76] In an empty directory, what is the output on your screen:
 

```
echo hi >.a ; echo ho >.b >.c ; ls *
```

  - a. `*`
  - b. an error message from `ls` saying `*` does not exist
  - c. no output
  - d. `hi ho`
  - e. `. . .a .b .c`
9. [36/76] 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 "$#$2"
```

 what is the output of that line?
  - a. `2c`
  - b. `2a`
  - c. `3b`
  - d. `3a`
  - e. `2b`
10. [37/76] In an empty directory, how many words are in file `c` after this command line: `touch a >b ; ls >c`
  - a. 1
  - b. 2
  - c. 3
  - d. 0
  - e. 4
11. [37/76] Which of these outputs an error message on Standard Error?
  - a. `echo 2>$1 'error'`
  - b. `echo 1>2 'error'`
  - c. `echo 2>&1 'error'`
  - d. `echo 1>$2 'error'`
  - e. `echo 1>&2 'error'`
12. [39/76] Remove the file named (including space and quotes):
 

```
It's "ugly"!
```

  - a. `rm It's\ "ugly"!`
  - b. `rm "It's" "ugly"!`
  - c. `rm 'It\'s\' "ugly"!'`
  - d. `rm "It's" 'ugly"!'`
  - e. `rm It\'s \'ugly\'`

13. [40/76] If a shell script named **foo** contains the line:  
`if [ '$3' = "$2" ] ; then echo SAME ; fi`  
 then which of the following command lines will always produce **SAME** as output?
- a. `./foo 1 '$3' 2`                      b. `./foo '$1' "$3" $2`  
 c. `./foo $1 '$2' $3`                    d. `./foo $1 $2 $3`  
 e. `./foo $3 "$2" $1`
14. [42/76] A Unix/Linux "tarball" is:
- a. a multi-file directory containing individual compressed files  
 b. a single-file that contains individual uncompressed files  
 c. a single compressed file containing one uncompressed file  
 d. a multi-file directory containing individual uncompressed files  
 e. a single-file that contains individual compressed files
15. [42/76] File **a** occupies one disk block. How many disk blocks are in use after this: `cp a b ; ln b c ; cp c d ; ln c e ; rm c`
- a. 4                      b. 3                      c. 2                      d. 0                      e. 1
16. [42/76] How many arguments are passed to the command by the shell:  
`$ <in cmd -a "-b c" '-d' >a b c d`
- a. 5                      b. 3                      c. 4                      d. 6                      e. 7
17. [42/76] If `/bin/foo` is a program that outputs **mom** and `/usr/bin/foo` is a program that outputs **dad**, what would be the output on your screen:  
`PATH=/dev:/usr/bin:/usr:/bin ; touch dad ; /bin/foo`
- a. **mom**  
 b. **bash: /dev/bin/foo: command not found**  
 c. **dad** followed by **mom**  
 d. **dad**  
 e. **bash: /bin/foo: command not found**
18. [42/76] which of the following statements is true about this line?  
`if [ "x" != "x" ] ; then echo NO ; fi`
- a. Command "[" is passed four arguments  
 b. The "if" keyword is always followed by "["  
 c. an "invalid number" error would result  
 d. "NO" would be printed  
 e. "fi" would cause a "command not found" error

19. [43/76] What would the following command do: **at 4pm**
- a. read commands from stdin to be run once at 4pm  
 b. run the user's **crontab** jobs every day at 4pm  
 c. issue an error message  
 d. run the user's **crontab** jobs once at 4pm  
 e. read commands from stdin to be run every day at 4pm
20. [44/76] File **foo** contains 9 lines, each of which is the one-digit line number of the line in the file (**1** through **9**), what is the output on your screen of this command:  
`cat foo foo | cat | tail -n 4 | head -n 1`
- a. 9                      b. 6                      c. 5                      d. 7                      e. 8
21. [45/76] How many lines are in file **out** after this:  
`date >wc >cat >out`
- a. 1 6 29                      b. 2                      c. 0 0 0  
 d. 1                              e. 0
22. [45/76] If my current directory is `/tmp`, which command copies the password file into existing directory `/tmp/dir` under the name **bar**?
- a. `cp ../../etc/passwd /dir/bar`  
 b. `cp ../tmp/./dir/./etc/passwd ./dir/./bar`  
 c. `cp ../../etc/passwd ../dir/bar`  
 d. `cp dir/../../etc/passwd dir/bar`  
 e. `cp ../dir/./etc/passwd ../tmp/dir/bar`
23. [45/76] 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 has the date and time in it  
 c. the personal crontab also has the userid in it  
 d. the system crontab has the date and time in it  
 e. the system crontab also has the userid in it
24. [46/76] Given a writable file named **ls**, what is the output on your screen:  
`echo hi >ls ; sort ls >ls ; cat ls`
- a. no output                      b. 1 1 2 ls                      c. 1 1 3 ls  
 d. 0 0 0 ls                      e. ls
25. [46/75] If a shell token with a GLOB pattern contains three slashes, how many slashes can be in each matched pathname?
- a. three or more                      b. two or three  
 c. zero, one, two, or three                      d. one, two, or three  
 e. exactly three

26. [46/76] In a directory containing one file named **dog**, what is the output on your screen after this: `1>/dev/null ls *`
- `ls: *: No such file or directory`
  - `bash: 1>/dev/null: command not found`
  - no output
  - `*`
  - `dog`
27. [46/76] The signal sent to a foreground process by typing the [Ctrl-C] key is:
- SIGINT**
  - SIGSTOP**
  - SIGTERM**
  - SIGKILL**
  - SIGHUP**
28. [46/76] Which **tar** command options create a gzip-style archive named **file**?
- `-cg file`
  - `-czf file`
  - `-cgf file`
  - `-gzip file`
  - `-cf file`
29. [47/76] File **a** contains 20 lines. File **b** contains 30 lines. How many lines are in file **out** after this:
- ```
cat a b >c ; head c >a ; sort a b c >out
```
- 30
  - 50
  - 100
  - 60
  - 90
30. [47/76] File **a** occupies one disk block. How many disk blocks are in use after this:
- ```
cp a b ; ln b c ; cp c a ; ln a e ; cp e c ; rm a b
```
- 4
  - 1
  - 0
  - 2
  - 3
31. [47/76] If **var=linus** then which one of the following **case** patterns will match this statement: `case "$var" in`
- `[linus] ) echo yes ; ;`
  - `'$var' ) echo yes ; ;`
  - `var ) echo yes ; ;`
  - `linu? ) echo yes ; ;`
  - `"l?nus" ) echo yes ; ;`
32. [47/76] 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 ; do
  s=$(( s + i + 1 ))
done
echo "$s"
```
- 6
  - 10
  - 3
  - 1 2 3
  - 9

33. [48/76] File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this:
- ```
ln a e ; ln b d ; ln d c ; cat a b e d >c
```
- 3
  - 2
  - 5
  - 4
  - 0
34. [48/76] If I have a directory named **x/y**, which action would increase its *link count* by exactly one?
- `ln x/y x/y2`
  - `mkdir x/y2`
  - `ln x x/y/z`
  - `ln -s x x/y2`
  - `mkdir x/y/z`
35. [48/76] In Linux system logging, processes:
- send messages to the **init** process that inherits orphan processes
  - copy logs from your **\$HOME** directory to the **/var/spool** directory
  - write log entries directly into the system log directory
  - write log files into each user's **\$HOME** directory
  - send messages to a central **rsyslog** program that writes log files
36. [48/76] Which **tar** options list a compressed archive without extracting it?
- `-ltf a.tar.gz`
  - `-xzf a.tar.gz`
  - `-tgz a.tar.gz`
  - `-tzf a.tar.gz`
  - `-lvf a.tar.gz`
37. [48/76] What would be the output of the following command line:
- ```
echo x y z | awk '{print $NF}'
```
- 3
  - NF**
  - z
  - \$NF**
  - x y z**
38. [48/76] If **a=123** and **b=456** then what is the output of the following command line: `if [ $a = $b ] ; then echo $a ; fi`
- 123
  - `test: a=123: integer expression expected`
  - `test: $a: string expression expected`
  - no output
  - `bash: 123: command not found`
39. [48/76] Search for **txt** inside file **foo** and print **OK** if found:
- `if fgrep 'txt' foo ; then echo OK ; fi`
  - `if test 'txt' = foo ; then echo OK ; fi`
  - `if [ test 'txt' foo ] ; then echo OK ; fi`
  - `if test 'txt' foo ; then echo OK ; fi`
  - `if [ fgrep 'txt' foo ] ; then echo OK ; fi`

40. [48/76] What system group of yours allows you to read the system **auth.log** file?
- a. **auth**                      b. **sys**                      c. **log**  
d. **adm**                      e. **gid**
41. [48/76] To append directory **/sbin** to your search path:
- a. **\$PATH=\$PATH:/sbin**                      b. **PATH=\$PATH:/sbin**  
c. **\$PATH=PATH:/sbin**                      d. **PATH=PATH:/sbin**  
e. **\$PATH+=PATH:/sbin**
42. [49/76] File **x** contains ten lines. File **y** contains twenty lines. How many lines are in file **cat** after this:
- ```
sort x y >z ; tail -n 5 y >y ; sort x y z >cat
```
- a. **40**                      b. **60**                      c. **0**                      d. **50**                      e. **45**
43. [49/76] Given an existing file named **cat**, what is the output on your screen: **echo xx >cat ; head cat >cat ; wc cat**
- a. **1 1 3 cat**                      b. no output                      c. **0 0 0 cat**  
d. **1 1 2 cat**                      e. **2 2 4 cat**
44. [49/76] In an empty directory, how many lines are in file **foo** after this:
- ```
ls nosuchfile . .. 2>foo
```
- a. **3**                      b. **0**                      c. **4**                      d. **2**                      e. **1**
45. [49/76] What value **umask** gives a new file permissions **----r--r--**?
- a. **623**                      b. **044**                      c. **644**                      d. **011**                      e. **022**
46. [49/76] If variable **var** might contain nothing (a null value - defined but empty), which command line correctly tests for this and prints **NULL**?
- a. **if [ "\$var" = '' ] ; then echo NULL ; fi**  
b. **if test "" -eq \$var ; then echo NULL ; fi**  
c. **if [ \$var = /dev/null ] ; then echo NULL ; fi**  
d. **if [ '\$var' = '' ] ; then echo NULL ; fi**  
e. **if test "\$var" = \* ; then echo NULL ; fi**
47. [50/75] File **a** occupies one disk block. How many disk blocks are in use after this: **cp a b ; ln b c ; cp c d ; cp c e**
- a. **4**                      b. **1**                      c. **5**                      d. **3**                      e. **2**
48. [50/76] File **a** occupies one disk block. How many disk blocks are in use after these commands:
- ```
ln a b ; ln b c ; cp c d ; ln c e ; rm a b c
```
- a. **2**                      b. **4**                      c. **3**                      d. **5**                      e. **1**

49. [50/76] 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 .**                      d. **ls /tmp/dir**  
e. **ls /tmp/dir/..**
50. [51/76] If you are in **/bin** and **ls -l** shows a symbolic link **bar -> ../dir/foo** then dereference the absolute path of **bar** with no symbolic links:
- a. **/bin/dir/foo**                      b. **/bin/dir/foo/bar**  
c. **/bar/../dir/foo**                      d. **/dir/foo**  
e. **/bin/bar/dir/foo**
51. [51/76] If you type the command **cat**, which **CTRL** key will send an **EOF** and take you back to the command prompt?
- a. **^D**                      b. **^R**                      c. **^E**                      d. **^U**                      e. **^C**
52. [51/76] What is in file **c** after this:
- ```
echo foo >a ; ln a b ; echo bar >>b ; ln a c ; rm a
```
- a. **foo**                      b. nothing (empty file)  
c. no such file (nonexistent)                      d. **bar**  
e. **foo** followed by **bar**
53. [52/76] Can files have the same inode number on different file systems?
- a. no: inode numbers only apply to directories, not files  
b. no: inode numbers are unique across all file systems  
c. no: you can't have inode numbers on more than one file system  
d. yes: inode numbers are only unique inside a file system  
e. yes: if the files are all names for the same inode
54. [52/76] Create a symbolic link under **/etc** named **bar** that has target **xyz**:
- a. **ln -s 'bar' /etc/xyz**  
b. **ln -s /etc/bar '/etc/xyz'**  
c. **ln -s "xyz" /etc/bar**  
d. **ln -s '/etc/xyz' 'bar'**  
e. **ln -s /etc/xyz '/etc/bar'**

55. [52/76] How many arguments and options are there to the command:  
**ls -al /etc**
- Two command line arguments, one of which contains two bundled options.
  - Two options, no arguments.
  - Two arguments, one of which is a single option and the other is a pathname.
  - A three-letter file name and an **/etc** switch option.
  - Two arguments, no options.
56. [52/76] If **/bin/bash** is a file name, which pathname always leads to the same file?
- /bin/bash/.**
  - ../bin/bash**
  - /../../bin/bash**
  - /bin/./bash**
  - ./bin/bash**
57. [52/76] If directory **a** contains only these five two-character names: **a?**, **11**, **?1**, **1\***, **.1**, then which command removes *only* the single two-character name **?1** from the directory?
- rm a/\*1**
  - rm a/?1**
  - rm a/??**
  - rm a/1\***
  - rm a/\??**
58. [52/75] 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 \*/.**
- 5 directory names
  - an error message because **\*/.** does not exist
  - 15 pathnames
  - \*/.**
  - no output
59. [52/76] Other than root, who can change the permissions of the following directory? **dr-xrwxrwx 17 bar foo 4096 Apr 15 16:40 .**
- only users in group **foo**
  - anyone except user **bar**
  - only user **bar**
  - user **bar** and any user in group **foo**
  - only **root** can change the permissions
60. [52/76] What is the link count of file **f** after these successful commands?  
**rm f ; touch f ; cp f x ; cp f f2**  
**ln f a ; ln x y ; ln a z ; ln z q**
- 2
  - 5
  - 6
  - 3
  - 4

61. [52/76] When an **at** job runs, the current working directory is set to:
- the directory with the name **/home**
  - the current directory that was in use when the **at** job was created
  - the directory with the name **/root**
  - the system ROOT directory
  - the HOME directory of the user who created the job
62. [52/76] Display all the names in the current directory that are exactly three digits long (and no other names):
- echo [???]**
  - echo [0-9][0-9][0-9]**
  - echo ???**
  - echo [0-9,0-9,0-9]**
  - echo [?][?][?]**
63. [52/76] If **work=payup** then which one of the following **case** lines will match this statement: **case "\$work" in**
- (\*yup echo yes ;;**
  - (\*payup\*) echo yes ;;**
  - [payup] ) echo yes ;;**
  - \* ) echo yes ;;**
  - ?payup? ) echo yes ;;**
64. [53/76] File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this: **cat a b >c ; cat a >>b ; sort c b >c a**
- 7
  - 12
  - 0
  - 5
  - 8
65. [53/76] File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this: **ln a d ; ln d c ; cat a b d >c**
- 3
  - 0
  - 4
  - 2
  - 5
66. [53/76] File **a** contains 3 lines. File **b** contains 4 lines. How many lines are output on your screen by this: **cat a ; echo a | echo b**
- 10
  - 2
  - 3 followed by 4
  - 4
  - 1 followed by 4
67. [53/76] In an empty directory, what is the output on your screen after this:  
**touch 1 2 .a .b ; echo .\***
- . . .a .b**
  - an error message from **echo** saying **.\*** does not exist
  - .a .b**
  - .\***
  - 1 2**
68. [53/76] To "throw away" (hide) standard error output of a command, use:
- cmd 2>/dev/sda1**
  - cmd 1>/dev/sda1**
  - cmd 2>&1**
  - cmd 2>/dev/null**
  - cmd 1>&2**

69. [53/76] What is the output of this in an empty directory:  
`touch .a .b .c ; echo [.]*`
- [.]\*
  - no output
  - . . . .a .b .c
  - an error message from `echo` saying [.]\* does not exist
  - .a .b .c
70. [53/76] Inside a shell script, which expands to the shell process ID?
- "\$?"
  - "\$\$"
  - "\$0"
  - "\$#"
  - "\$\*"
71. [54/76] A `crontab` time specification of `6 0 * * *` means at:
- 6:00am every business day and Saturday
  - 12:06am every day
  - 6:00am every business day
  - 6:00am every day
  - 12:06am every business day
72. [54/76] Which command searches for the text `other-writable` inside the text files of last term's course notes?
- `fgrep 'other-writable' oldnotes/*.txt`
  - `find oldnotes/*.txt -name 'other-writable'`
  - `find oldnotes/*.txt -name '*other-writable*'`
  - `fgrep oldnotes/*.txt 'other-writable'`
  - `find -name 'other-writable' oldnotes/*.txt`
73. [55/76] Given this `ls -il` long listing:  
`12 drwxr-xr-x 34 me me 56 Jan 1 1:00 dir`  
 How many subdirectories lie immediately under `dir`?
- 56
  - 54
  - 34
  - 32
  - 10
74. [55/76] If `/bin/foo` is a program that outputs `mom` and `/usr/bin/foo` is a program that outputs `dad` what would be the output on your screen of these two commands:  
`PATH=/etc:/bin/foo:/usr/bin/foo:/usr ; foo`
- `dad`
  - `mom`
  - `mom` followed by `dad`
  - `dad` followed by `mom`
  - `bash: foo: command not found`

75. [55/76] If directory `dir` contains only these four three-character file names: `.on`, `.tw`, `.th`, `.f.`, then what is the output on your screen of this: `echo dir/*`
- `dir/. dir/.. dir/.on dir/.tw dir/.th dir/.f.`
  - no output
  - `dir/.on dir/.tw dir/.th`
  - `dir/*`
  - `dir/.f.`
76. [55/76] In a directory containing one file named `dog`, what is the output on your screen after this: `2>/dev/null ls nosuchfile`
- `ls: nosuchfile: No such file or directory`
  - no output
  - `nosuchfile`
  - `dog`
  - `bash: 2>/dev/null: command not found`
77. [55/76] If the line, `exit 3` is executed in a shell script, what is the result?
- termination after sleeping for 3 seconds
  - termination with an exit status of 3
  - an invalid argument error message
  - the script breaks out of up to 3 levels of loops
  - termination with an exit status of 0
78. [56/76] 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: 500 File: 100
  - Directory: 600 File: 700
  - Directory: 400 File: 400
  - Directory: 200 File: 200
  - Directory: 100 File: 200
79. [56/76] How do I search for the string `foo` in the text display output from the `man` command?
- `search foo`
  - `@foo`
  - `/foo`
  - select "Search" in the menu
  - `find foo`

80. [56/76] If `/bin/bat` is a program that outputs `foo` and `/usr/bin/bat` is a program that outputs `bar` what would be the output on your screen of these two commands:  
`PATH=/etc:/usr:/usr/bin:/bin ; bat`
- `foo`
  - `bar` followed by `foo`
  - `bar`
  - `bash: bat: command not found`
  - `foo` followed by `bar`
81. [56/76] In an empty directory, what is output on your screen by:  
`mkdir -p 1/2/3 a/b/c ; mv 1/2/3 a/b ; find . -name 3`
- `./a/1/2/3`
  - `./a/b/1/2/3`
  - `./a/b/3`
  - `./a/b/c/1/2/3`
  - `./a/b/2/3`
82. [56/76] The **minimum** permissions you need to copy a file `foo` from directory `a` to directory `b` are:
- `wx` on `a`, `wx` on `b`, `rw` on `foo`
  - `wx` on `a`, `wx` on `b`, none on `foo`
  - `x` on `a`, `wx` on `b`, `r` on `foo`
  - `rx` on `a`, `wx` on `b`, `w` on `foo`
  - `rw` on `a`, `wx` on `b`, none on `foo`
83. [56/76] Inside a shell script, which expands to the name of the script itself?
- `"$*"`
  - `"$?"`
  - `"$0"`
  - `"$#"`
  - `"$@"`
84. [57/76] How many arguments are passed to the command by the shell:  
`$ echo " 1 '2 3' 4 " 5 6 ' 7 "8 '9 >out`
- 7
  - 3
  - 6
  - 5
  - 4
85. [57/76] How many lines are in the file `bar` after this:  
`echo hi >x ; echo ho >>x ; cat x x >bar`
- 1
  - 4
  - 0
  - 2
  - 6
86. [57/76] Display all the non-hidden names in the current directory that contain the letter `a` (and no other names):
- `echo *a`
  - `echo *a*`
  - `echo a*`
  - `echo ?a?`
  - `echo [a]`
87. [57/76] A pathname that is *not* an absolute pathname (after all shell expansions):
- `././dir`
  - `/dir`
  - `$HOME/dir`
  - `./dir`
  - `~/dir`

88. [57/75] Which `PATH` setting is from the Standard Script Header in this course?
- `PATH=/sh/bin:/usr/sh/bin`
  - `PATH=/bin:/usr/bin`
  - `PATH=/bin/sh:/usr/bin/sh`
  - `PATH=/bind:/usr/bind`
  - `PATH=/bin/sh:/sh/usr/bin`
89. [57/76] Which command lists your pending `at` jobs?
- `atq`
  - `atqu`
  - `at -q`
  - `at -queue`
  - `atqueue`
90. [58/76] Can user `emay` in group `greenpart` append to `./foo`?  
`dr-xr-xr-x 2 root greenpart 4096 Oct 7 14:00 .`  
`-r-xrwxrwx 1 emay greenpart 123 Oct 4 14:05 foo`
- No, because `emay` has no write permissions on `foo`
  - No, because `emay` has no write permission on the directory
  - Yes, because `emay` owns `foo`
  - No, because execute permissions are not set for `emay` on `foo`
  - No, because the directory is not accessible to `emay`
91. [58/76] File `foo` contains 9 lines, each of which is the one-digit line number of the line in the file (1 through 9), what is the output on your screen of this command:  
`cat foo foo | sort | tail -n 4 | head -n 1`
- 6
  - 8
  - 1
  - 4
  - no output
92. [58/75] Given the pathname `a/b/c`, the *basename* of this pathname is:
- `c`
  - `b/c`
  - `a`
  - `a/b`
  - `b`
93. [58/76] If `foo` is a sub-directory that contains only the file `bar`, what happens after this: `cp foo/bar ./foo/../me`
- the directory `foo` is now empty
  - there is a second copy of the file `bar` in directory `foo`
  - the directory `foo` now contains only a file named `me`
  - the command fails because the name `./foo/../me` does not exist
  - there is a second copy of the file `bar` in the file named `me`
94. [58/76] In an empty directory, how many arguments are passed to the `cat` command in this: `date >a1 ; touch a2 ba ca ; cat a*`
- 2
  - 3
  - 1
  - none
  - 4

95. [58/76] The **minimum** permissions you need to move a file **foo** from directory **a** to directory **b** are:
- wx** on **a**, **wx** on **b**, **w** on **foo**
  - wx** on **a**, **wx** on **b**, **r** on **foo**
  - rw** on **a**, **wx** on **b**, **rw** on **foo**
  - wx** on **a**, **wx** on **b**, none on **foo**
  - rw** on **a**, **wx** on **b**, none on **foo**
96. [58/76] What is the output on your screen after this:
- ```
echo one >x ; ln x y ; echo two >>y ; sort x
```
- one**
  - two** followed by **one**
  - two**
  - no output
  - one** followed by **two**
97. [58/76] What is true about this output from **ls -ild foo bar**
- ```
15 ----rwxrwx 2 me me 3 Jan 1 1:00 foo
15 d---rwxrwx 2 me me 3 Jan 1 1:00 bar
```
- foo** and **bar** are names for the same thing
  - foo** and **bar** each have three names (six names total)
  - foo** and **bar** are two of three names for the same thing
  - foo** and **bar** are names for different things
  - this output is not possible
98. [58/76] Which command displays all processes in a full wide listing?
- ps -all -wide**
  - ps -any -wide**
  - ps -full**
  - ps laxwx**
  - ps zxvf**
99. [59/76] Can user **emay** in group **greenpart** append to **./foo**?
- ```
drwxrw-rwx 2 root greenpart 4096 Oct 7 14:00 .
-rw-rw-r-- 1 emay greenpart 123 Oct 4 14:05 foo
```
- Yes, because **emay** has write permissions on **foo**
  - Yes, because **emay** owns **foo**
  - Yes, because **greenpart** has write permissions on **foo**
  - No, because execute permissions are not set for **emay** on **foo**
  - No, because the directory is not accessible to **emay**
100. [59/76] Can user **emay** in group **greenpart** copy **./foo** to **bar**?
- ```
drwxrw-r-x 2 root greenpart 4096 Oct 7 14:00 .
-rwx-wx-wx 1 emay greenpart 123 Oct 4 14:05 foo
```
- Yes, because **emay** has write permissions on **foo**
  - Yes, because **emay** owns **foo**
  - No, because the directory is not accessible to **emay**
  - No, because the directory has no write permissions for others
  - No, because **foo** has no read permissions for **emay**

101. [59/76] 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**?
- ln -s /a/foo /b/new**
  - ln /b/new /a/foo**
  - ln /a/foo /b/new**
  - ln -s /b/new /a/foo**
  - ln /a/new /b/foo**
102. [59/76] If my current directory is **/etc**, which of these pathnames is equivalent to the file name **/etc/passwd**?
- passwd/.**
  - ./etc/passwd**
  - ../etc/passwd/.**
  - ../../etc/passwd**
  - /root/etc/passwd**
103. [59/76] In an empty directory, what is output on your screen by:
- ```
mkdir -p 1/2/3 a/b/c ; mv 1/2 a/b ; find . -name 3
```
- ./a/b/1/2**
  - ./a/b/3**
  - ./a/1/2**
  - ./1/2/3**
  - ./a/b/2/3**
104. [59/75] 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
```
- pat** can rename the file
  - bob** can access and write on the file
  - pat** can access and write on the file
  - bob** can list names in the directory
  - bob** can create a new file in the directory
105. [59/76] Which command usually goes in your **.bash\_profile** file?
- ./bashrc source**
  - ./bashrc**
  - source ./bashrc**
  - source ./bash\_profile**
  - ./bash\_profile source**
106. [60/75] Can user **emay** in group **greenpart** append to **./foo**?
- ```
drwx---xwx 2 root greenpart 4096 Oct 7 14:00 .
-rw----- 1 emay greenpart 123 Oct 4 14:05 foo
```
- No, because execute permissions are not set for **emay** on **foo**
  - No, because the directory is not accessible to **emay**
  - No, because **greenpart** has no write permissions on **foo**
  - Yes, because **emay** has write permissions on **foo**
  - Yes, because **emay** owns **foo**

107. [60/76] What displays on your screen given this command:  
`date >date ; pwd >pwd ; head date | tail pwd`
- nothing displays because **tail** ignores the pipe
  - tail** reads the pipe and the **pwd** and displays both together
  - only the **pwd** file displays because **tail** ignores the pipe
  - only the **date** file displays because **tail** ignores the pipe
  - head** displays the **date** and **tail** displays the **pwd**
108. [60/76] What is the link count of directory **a** after these successful commands?  
`mkdir a ; mkdir a/b ; mkdir a/c ; mkdir a/b/c`
- 1
  - 4
  - 3
  - 5
  - 2
109. [60/76] What permissions are given to **newdir** after this:  
`umask 745 ; mkdir newdir`
- rxwx-wx-w-**
  - w--w-**
  - wx-w-**
  - rw-r--r--**
  - rxwxr--r-x**
110. [60/76] In a shell **case** structure, the **case** segment that will GLOB match the text **x**, **y**, or **z**, is coded as
- x\y\z )**
  - x|y|z )**
  - x/y/z )**
  - x,y,z )**
  - x:y:z )**
111. [61/76] Can user **emay** in group **greenpart** rename **./foo** to **bar**?  

```
d-----wx--- 2 root greenpart 4096 Oct 7 14:00 .
----- 1 emay greenpart 123 Oct 4 14:05 foo
```
- Yes, because **emay** owns **foo**
  - No, because the directory has no permissions for other users
  - No, because **emay** cannot read the directory
  - No, because **emay** has no permissions on **foo**
  - Yes, because **emay**'s group matches the group writable directory
112. [61/76] If a shell GLOB pattern fails to match anything, what happens by default? The shell:
- gives an error message and does not execute
  - removes the pattern and passes nothing
  - passes the pattern unchanged to the command
  - returns the closest match to the pattern
  - gives a warning message but continues

113. [61/76] If **foo** is a sub-directory that contains only the file **bar**, what happens after this: `mv ./foo/bar foo/./me`
- there is a second copy of the file **bar** in the file named **me**
  - the directory **foo** is now empty
  - the command fails because the name **me** does not exist
  - the command fails because the name **foo/./me** does not exist
  - the directory **foo** now contains only a file named **me**
114. [61/76] The **minimum** permissions you need to append to a file **foo** in directory **a** are:
- x** on **a**, **w** on **foo**
  - wx** on **a**, none on **foo**
  - rxw** on **a**, none on **foo**
  - wx** on **a**, **w** on **foo**
  - rxw** on **a**, **rw** on **foo**
115. [61/76] The **minimum** permissions you need to delete a file **foo** from directory **a** are:
- wx** on **a**, none on **foo**
  - wx** on **a**, **r** on **foo**
  - rxw** on **a**, **w** on **foo**
  - wx** on **a**, **w** on **foo**
  - rxw** on **a**, **rw** on **foo**
116. [61/75] 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
```
- bob** can access and write on the file
  - pat** can rename the file
  - bob** can list names in the directory
  - bob** can rename the file
  - pat** can create a new file in the directory
117. [61/76] When a personal **crontab** job runs, the current working directory is set to:
- the system **ROOT** directory
  - the current directory that was in use when the **crontab** job was created
  - the **HOME** directory of the user who created the job
  - the directory with the name **/home**
  - the directory with the name **/root**
118. [62/76] File **a** contains 2 lines. File **b** contains 3 lines. How many lines are output on your screen by this: `cat b | sort a`
- 3 followed by 2
  - 5
  - 2
  - 2 followed by 3
  - 3
119. [62/76] Give the minimum number of directories in this pathname:  
**/a/b/c**
- 3
  - 2
  - 4
  - 5
  - 1

120. [62/76] The **minimum** permissions you need to link a file **foo** from directory **a** to directory **b** are:
- rw**x on **a**, **w**x on **b**, **rw** on **foo**
  - rw**x on **a**, **w**x on **b**, none on **foo**
  - w**x on **a**, **w**x on **b**, **w** on **foo**
  - x** on **a**, **w**x on **b**, none on **foo**
  - w**x on **a**, **w**x on **b**, **r** on **foo**
121. [62/76] 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
```
- pat** can rename the file
  - bob** can access and write on the file
  - bob** can rename the file
  - pat** can create a new file in the directory
  - bob** can list names in the directory
122. [62/76] 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 rename the file
  - bob** can rename the file
  - bob** can access and write on the file
  - pat** can create a new file in the directory
  - bob** can list names in the directory
123. [62/75] 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
```
- pat** can rename the file
  - pat** can access and write on 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
124. [62/75] Output only lines 5-10 of the file named **foo**:
- head -n 5 foo | tail -n 10**
  - tail -n 15 foo | head -n 5**
  - tail -n 10 foo | head -n 6**
  - head -n 15 foo | tail -n 5**
  - head -n 10 foo | tail -n 6**

125. [63/76] If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
- ```
111 -rw-r--r-- 2 me me 100 Jan 1 1:00 a
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 b
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 c
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 d
```
- 0
  - 10
  - 3
  - 1
  - 2
126. [63/75] If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:
- ```
111 -rw-r--r-- 2 me me 100 Jan 1 1:00 a
111 -rw-r--r-- 2 me me 100 Jan 1 1:00 b
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 c
222 -rw-r--r-- 3 me me 100 Jan 1 1:00 d
```
- 3
  - 4
  - 2
  - 1
  - 8
127. [63/76] In a manual page **SYNOPSIS** section, ellipsis (three dots) (**...**) mean:
- a hidden directory
  - no special meaning
  - something that is optional
  - something that is repeated
  - the parent directory
128. [63/76] 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
```
- pat** can rename 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
  - bob** can access and write on the file
129. [63/76] 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
```
- bob** can list names in the directory
  - 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
130. [63/76] What command can you use to delete a directory that isn't empty?
- deltree -r dir**
  - rd -r dir**
  - del -r dir**
  - rm -r dir**
  - rmdir -r dir**

131. [63/76] What command displays the kernel ring buffer of log messages:  
 a. **showall**                      b. **ps lxww**                      c. **dmesg**  
 d. **psmine**                      e. **crontab**
132. [63/76] Which **crontab** time specification executes at **13:54** every day?  
 a. **13 54 \* \* \***                      b. **13 \* \* \* 54**  
 c. **\* \* \* 54 13**                      d. **\* \* \* 13 54**  
 e. **54 13 \* \* \***
133. [63/75] Which option to **ls** displays the directory itself and not its contents?  
 a. **-l**                      b. **-i**                      c. **-d**                      d. **-a**                      e. **-R**
134. [64/76] Dereference the following symlink **bar** into its equivalent absolute path: **ln -s ../../a/./b/./b/./foo /tmp/a/b/bar**  
 a. **/tmp/foo**                      b. **/tmp/a/b/bar**  
 c. **/tmp/a/b/foo**                      d. **/tmp/b/bar**  
 e. **/tmp/b/foo**
135. [64/76] In an empty directory, what is output on your screen by:  
**mkdir -p 1/2/3 a/b/c ; mv 1/2 a/b/c ; find . -name 3**  
 a. **./a/b/c/2/3**                      b. **./a/b/c/3**  
 c. **./a/b/c/1/2**                      d. **./a/b/1/2**  
 e. **./1/2/3**
136. [64/76] 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. **pat** can access and write on the file  
 b. **bob** can list names in the directory  
 c. **pat** can rename the file  
 d. **bob** can access and write on the file  
 e. **bob** can create a new file in the directory
137. [64/75] 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 ' ef " g h "**  
 how many times will the loop iterate?  
 a. 6 iterations                      b. 4 iterations                      c. 3 iterations  
 d. 7 iterations                      e. 5 iteration
138. [64/76] Which command displays a sum total of disk blocks?  
 a. **ls -d dir**                      b. **df -t dir**                      c. **ls -s dir**  
 d. **du -s dir**                      e. **tot -d dir**

139. [65/76] 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: **500**                      b. Directory: **300** File: **500**  
 c. Directory: **300** File: **200**                      d. Directory: **500** File: **500**  
 e. Directory: **100** File: **300**
140. [65/76] In a manual page **SYNOPSIS** section, square brackets (**[ ]**) mean:  
 a. an arithmetic expression  
 b. something that is optional  
 c. something that is repeated  
 d. no special meaning  
 e. a GLOB pattern matching a list
141. [65/76] 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 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
142. [65/75] User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.  
**d-wxr-xrw- 2 bob pgg 60 Jan 1 1:00 foo**  
**-r-xrwxr-x 1 bob bg1 0 Jan 1 1:00 foo/bar**  
 a. **pat** can access and write on the file  
 b. **bob** can list names in the directory  
 c. **pat** can rename the file  
 d. **bob** can access and write on the file  
 e. **bob** can create a new file in the directory
143. [65/76] User **bob** is in groups **bg1** and **bg2**. User **pat** is in group **pgg**.  
**d--x-----x 2 pat pgg 60 Jan 1 1:00 foo**  
**-r-xrwx-w- 1 bob bg1 0 Jan 1 1:00 foo/bar**  
 a. **pat** can 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

144. [65/76] 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`
- pat** can access and write on the file
  - 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
145. [65/76] What would be the output of the following command line:  
`echo a b c d e | awk '{print $3}'`
- c d e**
  - c**
  - a b c**
  - \$3**
  - 3**
146. [65/76] Which command line correctly compares the two numbers and prints **OK**?
- `if 4 > 3 ; then echo OK ; fi`
  - `if 4 >> 3 ; then echo OK ; fi`
  - `if [ 4 > 3 ] ; then echo OK ; fi`
  - `if $4 -gt $3 ; then echo OK ; fi`
  - `if [ 4 -gt 3 ] ; then echo OK ; fi`
147. [65/76] A shell script first line that would run the program **sh** from the **bin** directory to read the script file using one option:
- `#!/sh/bin -u`
  - `#!/bin/sh -u`
  - `#!/bin/sh -u`
  - `#!/sh/bin -u`
  - `#!/bin/sh -u`
148. [66/76] Can user **emay** in group **greenpart** remove `./foo`?  
`d----wx--- 2 root greenpart 4096 Oct 7 14:00 .`  
`----- 1 emay greenpart 123 Oct 4 14:05 foo`
- No, because the directory has no permissions for other users
  - No, because **emay** has no permissions on **foo**
  - No, because the directory is not accessible to **emay**
  - Yes, because **emay**'s group matches the group writable directory
  - Yes, because **emay** owns **foo**
149. [66/76] How many arguments does the shell pass to this **echo** command:  
`$ echo one two three >four five`
- 3**
  - 2**
  - 5**
  - 4**
  - 1**

150. [66/75] To have a user-defined alias in all your **bash** shells:
- define the alias in the file `.bash_history`
  - create the alias and then type "save" to save it
  - define the alias in the file `.bashrc`
  - create the alias and then type "newalias" to save it
  - define the alias in the file `/etc/alias`
151. [66/76] In an empty directory, what is output on your screen by:  
`mkdir -p 1/2/3 a/b/c ; mv 1 a/b ; find . -name 3`
- `./a/b/1/2/3`
  - `./a/b/c/1/2`
  - `./a/b/1`
  - `./a/1`
  - `./a/b/c/1/2/3`
152. [66/76] Under what directory are system log files usually stored?
- `/log/var`
  - `/usr/bin`
  - `/etc/log`
  - `/etc/passwd`
  - `/var/log`
153. [66/76] 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`
- bob** can list names in the directory
  - bob** can access and write on the file
  - pat** can access and write on the file
  - pat** can rename the file
  - bob** can create a new file in the directory
154. [66/76] 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`
- bob** can list names in the directory
  - 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
155. [66/76] 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`
- pat** 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
  - bob** can access and write on the file

156. [66/76] 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`
- 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 access and write on the file
  - pat** can rename the file
157. [66/76] A shell script is executed as follows:  
`./script "a b c" "d e" f`  
 Inside the script is the line: `echo "$2"`  
 What is the output on your screen from this line?
- "b" b. "\$2" c. b
  - d e e. \$2
158. [66/76] Inside a shell script, which expands to the number of script arguments?
- "\$\*" b. "\$?" c. "\$#"
  - "\$\$" e. "\$0"
159. [67/76] Can user **emay** in group **greenpart** remove `./foo`?  
`drwxr-xrwx 2 root greenpart 4096 Oct 7 14:00 .`  
`-rwxrwxrwx 1 emay greenpart 123 Oct 4 14:05 foo`
- No, because the directory is not accessible to **emay**
  - No, because the directory has no write permissions for **emay**
  - Yes, because **emay** has full permissions on **foo**
  - Yes, because **emay** owns **foo**
  - Yes, because **emay** matches the writable other permissions
160. [67/76] If you type the command `sleep 60`, which **CTRL** key will **interrupt** it and take you back to the command prompt?
- ^R** b. **^I** c. **^U** d. **^D** e. **^C**
161. [67/76] Which of the following signals is strongest (cannot be handled or ignored)?
- SIGKILL** b. **SIGTERM** c. **SIGHUP**
  - SIGSUSP** e. **SIGINT**

162. [67/75] 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`
- bob** can list names in the directory
  - pat** can create a new file in the directory
  - pat** can rename the file
  - bob** can create a new file in the directory
  - bob** can access and write on the file
163. [67/76] What value **umask** gives a new directory permissions `rw--w---x`?
- 421 b. 432 c. 211 d. 621 e. 156
164. [68/76] If **cow** is a sub-directory that contains only the file **dog**, what happens after this: `mv cow/dog cow/././cat`
- the directory **cow** is now empty
  - the command fails because the name **cat** does not exist
  - the command fails because the name `cow/././cat` does not exist
  - there is a second copy of the file **dog** in the file named **cat**
  - the directory **cow** now contains only a file named **cat**
165. [68/76] If **foo** is a sub-directory that contains only the file **bar**, what happens after this: `mv foo/./bar foo/././me`
- the command fails because the name **me** does not exist
  - the command fails because the name `foo/./bar` does not exist
  - there is a second copy of the file **bar** in the file named **me**
  - the directory **foo** is now empty
  - the directory **foo** now contains only a file named **me**
166. [68/76] 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`
- 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
  - pat** can access and write on the file



179. [71/75] **Did you read all the words of the test instructions on page one?**

- a. **mis** (*sey - eseugutrop*)                      b. **tak** (*yes - polish*)  
c. **jes** (*yes - esperanto*)                      d. **sej** (*sey - otnarepse*)  
e. **sim** (*yes - portuguese*)

180. [73/76] How do you execute the program **foo** in the current directory?

- a. **/foo**                      b. **\$HOME/foo**                      c. **../foo**  
d. **foo/.**                      e. **./foo**

*This page intentionally left blank.*