Evaluation: 110 Questions **Name:** _____

Important Instructions

1. Read all the instructions and both sides of all pages.
2. Manage your time when answering questions on this test. Answer the questions you know, first.

Multiple Choice - 110 Questions

This is a practice test containing practice questions. The real test will contain many questions similar to these. Knowing the concepts behind the questions is necessary; memorizing these answers won't help.

The real test may or may not have the same number of questions. The real test may have some questions unlike anything given here.

(Office use only: 44 61 84 99 60 56 28 77 40 98 14 21 109 90 24 89 20 87 79 67 108 29 64 110 16 82 9 17 32 45 8 37 88 30 83 12 103 104 15 107 51 65 13 94 52 75 35 69 78 71 53 18 73 97 38 3 41 4 23 7 48 39 43 1 5 47 102 54 46 74 36 57 66 92 76 100 68 95 42 49 62 70 96 63 6 31 26 101 85 2 58 106 55 25 93 80 34 33 22 59 50 27 19 81 11 10 105 72 91 86)

1. If a **bash** shell script named **foo** contains the line:
   ```
   if [ "$1" = '$2' ] ; then echo SAME ; fi
   ```
   then which of the following command lines will produce **SAME** as output?
   a. `./foo "bar" 'bar'`
   b. `./foo bar bar`
   c. `./foo $2 $2`
   d. `./foo "$1" '$2'`
   e. `./foo '$2' bar`

2. What is the **bash** output of this command sequence:
   ```
   false && echo "Hello     There"
   ```
   a. `"Hello     There"`
   b. `HelloThere`
   c. no output
   d. `Hello     There`
   e. `Hello There`

3. If directory **/dir** contains these three four-character file names: **.123**, **.124**, **.???**, then what is the output of the following **bash** shell command line: **echo /dir/????**
   a. no output
   b. `/dir/.123 /dir/.124 /dir/.???`
   c. `echo: /dir/????: No such file or directory`
   d. `/dir/.123 /dir/.124`
   e. `/dir/????`

4. Which command line displays the contents of the Unix **passwd** file one page at a time?
   a. `cat /etc/passwd >less`
   b. `less </etc/passwd`
   c. `/etc/passwd | less`
   d. `/etc/passwd >less`
   e. `less | /etc/passwd`

5. What is the **bash** shell output of this command sequence:
   ```
   true && echo space     junk $?
   ```
   a. `space     junk ?`
   b. `space junk ?`
   c. no output
   d. `space junk 0`
   e. `space     junk 1`

6. If **x=8** and **y=9** then which of the following **bash** command lines outputs only the word **foobar** (and nothing else)?
   a. `[ x -ne y ] && echo foobar`
   b. `[x -ne y] || echo foobar`
   c. `[x!=x] || echo foobar`
   d. `[ x = x ] && echo foobar`
   e. `[!x = y] && echo foobar`

7. What is the output of the following sequence of **bash** commands:
   ```
   x=0 ; test $x ; echo $?
   ```
   a. `0`
   b. no output
   c. `test: $x: integer expression expected`
   d. `1`
   e. the number 0 or 1 followed by another 0 or 1 on a new line

8. If file **/a** contains 20 lines, and file **/b** contains 30 lines, then how many lines are in file **/c** after this sequence of shell commands:
   ```
   sort /a /b >/c ; cat /a >>/b ; sort /c /b /a
   >/c
   ```
   a. `80`
   b. `120`
   c. `50`
   d. `70`
   e. no lines (empty file)

9. Which **bash** command sequence correctly searches for the string **foo** and then prints **YES** if it is found inside the group file?
   a. **if [test foo /etc/group] ; then echo YES ; fi**
   b. **if test foo = /etc/group ; then echo YES ; fi**
   c. **if test foo /etc/group ; then echo YES ; fi**
   d. **if grep foo /etc/group ; then echo YES ; fi**
   e. **if [grep foo /etc/group] ; then echo YES ; fi**

10. If my current working directory is **/home**, and my home directory is **/home/me**, which of the of the following commands copies the Unix password file into my home directory under the name **foo**?
   a. **cp ./me/../etc/passwd ../home/me/foo**
   b. **cp me/../../etc/passwd me/foo**
   c. **cp ../../etc/passwd /me/foo**
   d. **cp ../home/me/../etc/passwd ./me/./foo**
   e. **cp ../etc/passwd ../me/foo**

11. If **x=pig** and **y=dog** then what is the output of the following sequence of **bash** commands: **if $x = $y ; then echo $y ; fi**
   a. **test: $x: integer expression expected**
   b. no output
   c. **test: pig: integer expression expected**
   d. **bash: pig: command not found**
   e. **dog**

12. What is the output of the following sequence of **bash** commands:
```
false && echo "foo     bar $?"
```
   a. **foo bar 1**
   b. **foo   bar 1**
   c. no output
   d. **foo   bar 0**
   e. **foo bar 0**

13. Given this long listing:
```
drwxr-xr-x 2048 root root 4096 Jan 20 14:22 /dir-
```
How many subdirectories lie immediately under **/dir**?
   a. there is not enough information shown to answer the question
   b. 4096
   c. 2046
   d. 2048
   e. 4094

14. Which command sequence below does *not* generate an error message from the last command in the sequence?
   a. **mkdir foo foo/bar ; rmdir foo**
   b. **mkdir one one/two ; rmdir one/two**
   c. **cat /etc/passwd > mail idallen@ncf.ca**
   d. **mkdir foo ; ln foo bar**
   e. **date >foo ; cp foo/. bar**

15. What is the output of the following sequence of **bash** commands:
```
x=cow ; y=dog ; touch $x ; test -z $x ; echo $?
```
   a. **0**
   b. no output
   c. the number 0 or 1 followed by another 0 or 1 on a new line
   d. **1**
   e. **test: $x: integer expression expected**

16. Which command sequence below outputs only lines 11-15 of the Unix password file?
   a. **head -10 /etc/passwd | tail -15 /etc/passwd**
   b. **head -15 /etc/passwd | tail -5 /etc/passwd**
   c. **tail -10 /etc/passwd | head -15 /etc/passwd**
   d. **tail -15 /etc/passwd | head -10**
   e. **head -15 /etc/passwd | tail -5**

17. What is the output of the following sequence of **bash** commands:
```
wc='one two' ; test wc = wc
```
   a. **test: too many arguments**
   b. **0**
   c. **1**
   d. no output
   e. **1   2   8 wc**

18. If the file **foo** contained the word **mom**, what would be the **bash** shell output of this two command sequence:
```
PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/ls foo
```
   a. **/bin/ls: foo: No such file or directory**
   b. **foo**
   c. no output
   d. **bash: /bin/ls: command not found**
   e. **mom**

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

20. How many arguments are passed to the command by the shell on this command line: **<cow cow "-x" -y '-z' >cow cow**
a. **2**
b. **3**
c. **4**
d. **5**
e. **6**

21. How can you ask the **bash** (Linux) shell to complete commands or file names for you?
a. You can type the first part of the command or file name and press the **ALT** key.
b. Type **[CONTROL]-[D]** and the shell will present a menu of commands.
c. You can type the first part of the command or file name and press the **TAB** key.
d. Type **[ALT]-[F2]** the shell will present a menu of commands.
e. Type **[CONTROL]-[ALT]-[DEL]** and the shell will present a menu of commands.

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

23. What is the output of this sequence of three shell commands:
```
umask 674 ; touch newfile ; ls -l newfile
```
a. **-rw-rw-r-- 1 me me 0 Feb 20 07:55 newfile**
b. **--w--wxr-x 1 me me 0 Feb 20 07:55 newfile**
c. **---x----wx 1 me me 0 Feb 20 07:55 newfile**
d. **--------w- 1 me me 0 Feb 20 07:55 newfile**
e. **-rw-rwxr-- 1 me me 0 Feb 20 07:55 newfile**

24. Select the correct **bash** shell order of command line processing:
a. aliases, redirection, variables, globs
b. aliases, globs, variables, redirection
c. aliases, variables, globs, redirection
d. aliases, variables, redirection, globs
e. redirection, aliases, globs, variables

25. Which **bash** command line below allows programs in the current directory to execute without preceding the names with **./**?
a. **$PATH=/usr/bin:./$HOME**
b. **PATH=/usr/bin/.:$HOME**
c. **PATH=./$HOME:/usr/bin**
d. **$PATH=.:$HOME:/usr/bin**
e. **PATH=/usr/bin:$HOME:.**

26. What is the **bash** shell output of this two command sequence:
```
PATH=/bin/ls:/bin/cat:/bin/sh ; cat nosuchfile
```
a. **bash: cat: command not found**
b. **bash: /bin/sh: No such file or directory**
c. **ls: /bin/cat: command not found**
d. **cat: nosuchfile: No such file or directory**
e. **bash: /bin/ls: command not found**

27. If **a=cow** and **b=dog** then what is the output of the following sequence of **bash** commands: **[$a = cow -a $b = cow] ; echo $?**
a. **0**
b. **1**
c. **test: $a: integer expression expected**
d. the number 1 or 0 followed by another 1 or 0 on a new line
e. no output

28. What is the output of the following sequence of **bash** commands:
```
echo wc >wc ; wc wc >wc ; cat wc
```
a. no output
b. **1 1 3 wc**
c. **1 1 2 wc**
d. **wc**
e. **0 0 0 wc**

29. Which of these statements is true?
a. The **cat food** command looks up the file name argument **food** in your **$PATH**.
b. If **/x** is an empty directory, **sort /x/*** produces an error message.
c. Only single quotes are strong enough to stop shell glob (wildcard) patterns from expanding.
d. Typing **./script** and **bash script** always give identical results.
e. If **/y** is an empty directory, **echo /y/*** produces an error message.

30. If **foo** is a script containing the line **TERM=vt100 ; export TERM**, what is the output of the following sequence of **bash** commands:
    TERM=linux ; ./foo ; echo $TERM
    a. **vt100**
    b. **TERM**
    c. **linux**
    d. **$TERM**
    e. **foo**

31. If **a=cow** and **b=dog** then what is the output of the following sequence of **bash** commands: **[$a = dog -o $b = cow] ; echo $?**
    a. the number 1 or 0 followed by another 1 or 0 on a new line
    b. no output
    c. **0**
    d. **1**
    e. **test: $a: integer expression expected**

32. A shell script named **foo** is executed as follows:
    ./foo a "b c d" e
    Inside the script is the line: **echo "$2"**
    What is the output from this line?
    a. **b c d**
    b. **b**
    c. a bash error message: unbound (undefined) variable
    d. **$2**
    e. **"b**

33. If **/bin/foo** is a program that outputs **hi** and **/usr/bin/foo** is a program that outputs **mom** what is the output of this shell command sequence: **PATH=/etc:/usr/bin:/bin ; foo**
    a. **mom** followed by **hi**
    b. **hi**
    c. **bash: foo: command not found**
    d. **mom**
    e. **hi** followed by **mom**

34. What is the output of the following sequence of **bash** commands:
    x=1 ; y=2 ; test $x -le $y ; echo $?
    a. **0**
    b. **1**
    c. **test: $x: integer expression expected**
    d. the number 0 or 1 followed by another 0 or 1 on a new line
    e. no output

35. Which of the following statements is true about this shell command line:
    >foo file bar haven
    a. The command **file** sees two arguments.
    b. The command **foo** sees only two arguments
    c. The command **foo** sees three arguments.
    d. The command **file** sees three arguments.
    e. Error: The command name is missing from the command line.

36. If **a=cow** and **b=dog** then what is the output of the following sequence of **bash** commands linked by **&&**:
    [$a = dog -o $b = dog] && echo $?
    a. **0**
    b. no output
    c. **test: $a: integer expression expected**
    d. **1**
    e. the number 1 or 0 followed by another 1 or 0 on a new line

37. If directory **dir** contains only these five two-character file names: **a?**, **11**, **?1**, **1***, **.1**, then which shell command below will remove *only* the single two-character name **?1** from the directory?
    a. **rm dir/\??**
    b. **rm dir/1***
    c. **rm dir/??**
    d. **rm dir/*1**
    e. **rm dir/?1**

38. Which of the following VI/VIM key sequences will move the entire line on which the cursor resides to after the line that follows it (i.e. it would move line 5 to be line 6 and line 6 would become line 5)?
    a. **DDP**
    b. **DDp**
    c. **:dp**
    d. **ddP**
    e. **ddp**

39. Which **bash** command line below allows programs in the current directory to execute without preceding the names with **./**?
    a. **PATH = /bin:$HOME:.**
    b. **PATH=/bin:$HOME:.**
    c. **PATH = ./$HOME:/bin**
    d. **$PATH=/bin:./$HOME**
    e. **$PATH=.:$HOME:/bin**

40. Which line below passes three *separate* arguments to the **sort** command when placed inside a shell script named **foo** invoked by the command line:
```
./foo  111  222  333
```
a. **sort "$#"**
b. **sort "$*"**
c. **sort "$? $? $?"**
d. **sort "$1 $2 $3"**
e. **sort "$@"**

41. How many arguments and options are there to the command:
```
ls -lid /p
```
a. Two arguments, one of which is a single option name and the other is a pathname.
b. Two command line arguments, one of which contains three options.
c. Two arguments, neither of which is an option.
d. Two arguments: A file name starting with a dash and a **/p** switch option argument.
e. Three arguments, one of which contains options and one is a pathname.

42. What is the output of this sequence of three shell commands:
```
umask 547 ; mkdir newdir ; ls -ld newdir
```
a. **d-w--wxrwx 1 me me 0 Feb 20 07:55 newdir**
b. **d-w--w---- 1 me me 0 Feb 20 07:55 newdir**
c. **dr--r--rw- 1 me me 0 Feb 20 07:55 newdir**
d. **dr-xr--rwx 1 me me 0 Feb 20 07:55 newdir**
e. **d-w--wx--- 1 me me 0 Feb 20 07:55 newdir**

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

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

45. How many arguments and options are there to the command:
```
wc -l <infile
```
a. Two arguments, one of which is a single option name and the other is a pathname.
b. One command line argument containing one option name.
c. Three arguments, one of which contains an option and one is a pathname.
d. Two arguments, neither of which is an option.
e. A file name starting with a dash and an **<infile** switch option argument.

46. Which of the following shell command lines displays all the names in the current directory that are exactly three letters (alphabetic) long (and nothing else)?
a. **echo [a,zA,Z][a,zA,Z][a,zA,Z]**
b. **echo [azAZ][azAZ][azAZ]**
c. **echo [0-89][01-9][0-45-9]**
d. **echo ???**
e. **echo [a-mn-zA-YZ][ab-zAB-YZ][za-yZA-Y]**

47. In an empty directory, what is the length of the longest file name created by the following **bash** shell two-command sequence:
```
ok='1 12 123 1234' ; touch '$ok'
```
a. 4 characters
b. 3 characters
c. 2 characters
d. 1 character
e. 13 characters

48. In an empty directory, what is the shell output of these three commands:
```
touch .1 .2 .3 11 12 ; b='.1* .2*' ; echo '$b'
```
a. **.1* .2***
b. **.1 .2**
c. **$b**
d. **11 .1 12 .2**
e. **'.1* .2*'**

49. If file **/a** contains 40 lines, and file **/b** contains 60 lines, then how many lines are output by this command:
```
sort /a /b | cat /a | cat /b
```
a. **100**
b. **60**
c. **160**
d. **200**
e. **40**

50. Given my directory **dir** and my file **dir/foo** owned by me, which permissions allow me to change or create new content (data) in the file **dir/foo** but not delete the file?
    a. Permissions **600** on directory **dir** and **700** on file **dir/foo**.
    b. Permissions **500** on directory **dir** and **600** on file **dir/foo**.
    c. Permissions **400** on directory **dir** and **400** on file **dir/foo**.
    d. Permissions **100** on directory **dir** and **100** on file **dir/foo**.
    e. Permissions **200** on directory **dir** and **200** on file **dir/foo**.

51. How many arguments and options are there to the command:
    ```
    cal -yj 1752
    ```
    a. Two arguments, neither of which is an option.
    b. A single numeric option and a three-letter file name.
    c. Three arguments, one of which contains options and one is a pathname.
    d. Two arguments, one of which is a single option and the other is a pathname.
    e. Two command line arguments, one of which contains two options.

52. What is the output of the following sequence of **bash** commands:
    ```
    echo wc >wc ; wc wc >wc ; sort wc
    ```
    a. **0 0 0 wc**
    b. **wc**
    c. no output
    d. **1 1 2 wc**
    e. **1 1 3 wc**

53. What is the link count of directory **dir** after this set of successful commands? **mkdir dir ; cd dir ; touch one ; mkdir two**
    a. 2
    b. 4
    c. 5
    d. 1
    e. 3

54. If **/etc/passwd** is a file name, which of the following pathnames always leads to the same file?
    a. **/etc/passwd/./.**
    b. **/etc/passwd/../..**
    c. **/etc/../../passwd**
    d. **././etc/passwd**
    e. **/./../etc/./passwd**

55. Which **bash** command sequence correctly searches for the **chars** and then prints **OK** if it is found inside the password file?
    a. **grep chars >/etc/passwd && echo OK**
    b. **grep chars </etc/passwd || echo OK**
    c. **grep chars /etc/passwd || echo OK**
    d. **grep chars >/etc/passwd || echo OK**
    e. **grep chars </etc/passwd && echo OK**

56. If **x=5** and **y=5**, which **bash** command sequence correctly compares the two numbers as equal and prints **OK**?
    a. **if [$x==$y] ; then echo OK ; fi**
    b. **if test $x -eq $y ; then echo OK ; fi**
    c. **if [x = y] ; then echo OK ; fi**
    d. **if test x -eq y ; then echo OK ; fi**
    e. **if (x == y) ; then echo OK ; fi**

57. If file **foo** contains nine lines, each of which is the one-digit line number of the line in the file (**1** through **9**), what is the output of this command:
    ```
    cat foo foo | cat | tail -4 | head -1
    ```
    a. **6**
    b. **5**
    c. **7**
    d. **9**
    e. **8**

58. If **cow=5** and **dog=5**, which **bash** command sequence correctly compares the two numbers as equal and prints **OK**?
    a. **if [cow = dog] ; then echo OK ; fi**
    b. **if test $cow -eq $dog ; then echo OK ; fi**
    c. **if test cow -eq dog ; then echo OK ; fi**
    d. **if (cow == dog) ; then echo OK ; fi**
    e. **if [$cow==$dog] ; then echo OK ; fi**

59. What is the output of the following sequence of **bash** commands:
    ```
    a=cow ; b=dog ; touch $a ; test -z $a ; echo $?
    ```
    a. no output
    b. **0**
    c. the number 1 or 0 followed by another 1 or 0 on a new line
    d. **1**
    e. **test: $a: integer expression expected**

60. If variable **bar** might contain nothing (a null value - defined but empty), which **bash** command sequence correctly tests for this and prints **YO**?
    a. `if [ "$bar" = * ] ; then echo YO ; fi`
    b. `if [ $bar -eq : ] ; then echo YO ; fi`
    c. `if [ "$bar" = "" ] ; then echo YO ; fi`
    d. `if [ $bar -eq "" ] ; then echo YO ; fi`
    e. `if [ ''$bar'' = '''' ] ; then echo YO ; fi`

61. What is the **bash** shell output of this two-command sequence if run in a directory containing 888 files with names that are all the numbers from **1** to **888** inclusive: `cow="*" ; echo '$cow'`
    a. `$cow`
    b. `'$cow'`
    c. the file names **1** through **888**
    d. the file names **1** through **888**, surrounded by quotes
    e. `*`

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

63. If a **bash** shell script named **foo** contains the line:
    `if [ '$1' = "$2" ] ; then echo SAME ; fi`
    then which of the following command lines will produce **SAME** as output?
    a. `./foo 1 "$1"`
    b. `./foo bar '$1'`
    c. `./foo 'bar' "bar"`
    d. `./foo bar 'bar'`
    e. `./foo $1 $1`

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

65. If variable **foo** might contain nothing (a null value - defined but empty), which **bash** command sequence correctly tests for this and prints **OK**?
    a. `if [ "$foo" = * ] ; then echo OK ; fi`
    b. `if [ ''$foo'' = '''' ] ; then echo OK ; fi`
    c. `if [ "$foo" = "" ] ; then echo OK ; fi`
    d. `if [ $foo -eq "" ] ; then echo OK ; fi`
    e. `if [ $foo -eq : ] ; then echo OK ; fi`

66. If **foo** is a script containing the line `TERM=new ; export TERM`, what is the output of the following sequence of **bash** commands that use **foo**:
    `TERM=bar ; ./foo ; echo $TERM`
    a. `new`
    b. `bar`
    c. `$TERM`
    d. `TERM`
    e. `foo`

67. What is the output of this sequence of three shell commands:
    `echo x >abc ; ls >abc abc ; wc abc`
    a. `1 1 4 abc`
    b. no output
    c. `0 0 0 abc`
    d. `1 1 3 abc`
    e. `1 1 2 abc`

68. If **a=1** and **b=2** then which of the following **bash** command lines outputs only the word **hi** (and nothing else)?
    a. `[ a = a ] && echo hi`
    b. `[a!=a] || echo hi`
    c. `[a -ne b] || echo hi`
    d. `[ a -ne b ] && echo hi`
    e. `[!a = b] && echo hi`

69. If **foo** is a script containing the line `TERM=linux ; export TERM`, what is the output of the following sequence of **bash** commands:
    `TERM=vt100 ; ./foo ; echo "$TERM"`
    a. `$TERM`
    b. `vt100`
    c. `foo`
    d. `TERM`
    e. `linux`

70. What is the link count of file **foo** after this set of successful commands?
```
rm foo ; touch foo ; ln foo bar
cp bar x ; ln x y ; ln bar z
```
a. 1
b. 4
c. 3
d. 5
e. 2

71. A shell script named **foo** is executed as follows:
```
./foo 1 "2 3 4" 5
```
Inside the script is the line: **echo "$2"**
What is the output from this line?
a. **$2**
b. **2 3 4**
c. **"2**
d. **"2 3 4"**
e. **2**

72. Which **bash** command sequence below always outputs just the date only if the first argument is either a file or a directory?
a. **if [-f || -d "$1"]; then date ; fi**
b. **if ["$1" -eq -f -o "$1" -eq -d]; then date ; fi**
c. **if ["-f $1" || "-d $1"]; then date ; fi**
d. **if [-f "$1" -o -d "$1"]; then date ; fi**
e. **if [-f -o -d "$1"]; then date ; fi**

73. How many arguments are passed to the command by the shell on this command line: **<bar bar -b "-a" '-r' >bar bar bar**
a. **2**
b. **6**
c. **4**
d. **5**
e. **3**

74. Which of these first lines will cause this executable file to be interpreted using the Bash shell?
a. **!/bin/bash**
b. **/bin/bash -u**
c. **#!/bin/bash**
d. **#/bin/bash**
e. **!#/bin/bash -u**

75. Which of the following shell command lines displays only the names in the current directory that are exactly three numeric digits long?
a. **echo [1-3][1-3][1-3]**
b. **echo [0-9][0-9][0-9]**
c. **echo ???**
d. **echo '0-9'0-9'0-9'**
e. **echo '[0-9]''[0-9]''[0-9]'**

76. Which of the command lines below can generate a non-empty file?
a. **sort -r /a/b >/a/b**
b. **tr abc ABC </a/b >/a/b**
c. **tail -5 /a/b >/a/b**
d. **grep -v /a/b /a/b >/a/b**
e. **ls /a/b >/a/b**

77. In an empty directory, what is the shell output of these three commands:
```
touch xx .x xy .y xz ; a='x* y*' ; echo "$a"
```
a. **x* y***
b. **xx xy**
c. **$a**
d. **xx xy xz y***
e. ***x *y**

78. Which line below is most likely to be the beginning of an error message?
a. **echo 2>&1 "... "**
b. **echo 1>&2 "... "**
c. **echo 2<$1 "... "**
d. **echo 2>$1 "... "**
e. **echo 1<&2 "... "**

79. If file **foo** contains nine lines, each of which is the one-digit line number of the line in the file (**1** through **9**), what is the output of this command:
```
cat foo foo | sort -r | head -5 | tail -1
```
a. **7**
b. **5**
c. **6**
d. **8**
e. **9**

80. What is the **bash** shell output of this two-command sequence if run in a directory containing 123 files with names that are all the numbers from **1** to **123** inclusive: **bat="*" ; echo "$bat"**
a. the file names **1** through **123**
b. the file names **1** through **123**, surrounded by quotes
c. ***
d. **$bat**
e. **"$bat"**

81. Given the following **bash** shell command line: **read a b c**, which user keyboard input line below will assign the text **b** to the shell variable named **b**?
    a. **a;b;c**
    b. **a,b,c**
    c. **a=a b=b c=c**
    d. **a:b:c**
    e. **a b c**

82. In an empty directory, what is the **bash** shell output of this three-command sequence:
    touch aa .a ab .b .c ; x='.a* .b*' ; echo '$x'
    a. **$x**
    b. **'.a* .b*'**
    c. **aa .a ab .b**
    d. **.a .b**
    e. **.a* .b***

83. Which line below puts the count of the number of lines in the password file into the variable **foo**?
    a. **foo=[wc /etc/passwd | echo $1]**
    b. **foo=$(cat -c /etc/passwd)**
    c. **foo=$(wc -l </etc/passwd)**
    d. **foo=[grep -c /etc/passwd]**
    e. **foo=[cat -l /etc/passwd]**

84. Given the following **bash** shell command line:
    read xx yy zz
    which user keyboard input line below will assign the text **22** to the shell variable named **yy**?
    a. **11;22;33**
    b. **11:22:33**
    c. **11 22 33**
    d. **xx=11 yy=22 zz=33**
    e. **11,22,33**

85. If variable **cow** might contain nothing (a null value - defined but empty), which **bash** command sequence correctly tests for this and prints **OK**?
    a. **if ["$cow" = *] ; then echo OK ; fi**
    b. **if [$cow -eq ""] ; then echo OK ; fi**
    c. **if [$cow -eq :] ; then echo OK ; fi**
    d. **if [''$cow'' = ''''] ; then echo OK ; fi**
    e. **if ["" = "$cow"] ; then echo OK ; fi**

86. What is the output of the following sequence of **bash** commands:
    false && echo "linux rocks $?"
    a. **linux rocks 0**
    b. **linux rocks 1**
    c. **linux rocks 0**
    d. no output
    e. **linux rocks 1**

87. What is the output of the following sequence of **bash** commands:
    cd /bin && echo "cd $(pwd)"
    a. **cd $(pwd)**
    b. no output
    c. **cd /bin**
    d. **bash: cd: /bin: No such file or directory**
    e. **cd 0pwd)**

88. Which of the following **bash PATH** statements makes the most sense?
    a. **PATH=/bin/ls:/etc:/usr/bin**
    b. **PATH=/bin:/bin/cat:/usr/bin**
    c. **PATH=/bin:/usr/bin:/etc**
    d. **PATH=/bin:/usr/bin:/etc/passwd**
    e. **PATH=/bin/sh:/usr/bin:/etc:/bin**

89. What is in the file named **file** after this command sequence:
    echo a >x ; echo b >>x ; mv x y >file
    a. **a**
    b. nothing - **file** is empty - no data
    c. no such file (nonexistent file)
    d. **b**
    e. **a** followed by **b**

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

91. Which **bash** command sequence below always outputs just the date only if the first argument is either readable or executable?
    a. **if [-r -o -x "$1"]; then date ; fi**
    b. **if ["$1" -eq -r -o "$1" -eq -x]; then date ; fi**
    c. **if ["-r $1" || "-x $1"]; then date ; fi**
    d. **if [-r "$1" -o -x "$1"]; then date ; fi**
    e. **if [-r || -x "$1"]; then date ; fi**

92. If **foo** were a file of text containing 50 different lines, what would be the output of this exact command line: **diff foo foo**
    a. no output
    b. the contents of file **foo** would be displayed
    c. an error message because **diff** doesn't allow the same file name twice
    d. several lines, which are the lines that are different between the two files
    e. an error message because **diff** only allows one file name

93. If **dog=12** and **cat=99** then which of the following **bash** command lines outputs only the word **hi** (and nothing else)?
    a. **[dog -ne cat] && echo hi**
    b. **[dog!=dog] || echo hi**
    c. **[dog -ne cat] || echo hi**
    d. **[dog = dog] && echo hi**
    e. **[!dog = cat] && echo hi**

94. What is the output of the following sequence of **bash** commands:
    cd /etc && echo "in $(pwd)"
    a. **in 0pwd)**
    b. **in /etc**
    c. **bash: cd: /etc: No such file or directory**
    d. **in $(pwd)**
    e. no output

95. Which Unix command sequence deletes a directory and everything inside it?
    a. **rmdir -all dir**
    b. **deltree -all dir**
    c. **rm -all dir**
    d. **rmdir -r dir**
    e. **rm -r dir**

96. What is the **bash** shell output of this two-command sequence:
    cd /home/alleni && echo "In $(pwd)"
    a. no output
    b. **"In $(pwd)"**
    c. **In 0pwd)**
    d. **In /home/alleni**
    e. **In $(pwd)**

97. In an empty directory, what is the length of the longest file name created by the following **bash** shell two-command sequence:
    x='1 12 123 1234' ; touch '$x'
    a. 2 characters
    b. 3 characters
    c. 13 characters
    d. 4 characters
    e. 1 character

98. Which of these statements is true?
    a. Either single or double quotes will stop shell glob (wildcard) patterns from expanding.
    b. The **ls dir** command looks up the directory argument **dir** in your **$PATH**.
    c. If **/q** is an empty directory, **echo /q/.*** produces an error message.
    d. If **/p** is an empty directory, **ls /p/.*** produces an error message.
    e. Typing **./script** and **bash script** always give identical results.

99. What is the output of the following sequence of **bash** commands:
    x=0 ; y=1 ; touch $x ; test ! -z $x ; echo $?
    a. **1**
    b. **0**
    c. the number 1 or 0 followed by another 1 or 0 on a new line
    d. **test: $x: integer expression expected**
    e. no output

100. What is the **bash** shell output of this command sequence:
    true && echo Hello There $?
    a. no output
    b. **Hello There ?**
    c. **Hello There ?**
    d. **Hello There 0**
    e. **Hello There 1**

101. What is the **bash** shell output of this two-command sequence if run in a directory containing 765 files with names that are all the numbers from **1** to **765** inclusive: **foo="*" ; echo $foo**
    a. **$foo**
    b. all the file names that start with an asterisk ('*****')
    c. an asterisk ('*****') and the file names **1** through **765**
    d. the file names **1** through **765**
    e. *****

102. What is the output of the following sequence of **bash** commands:
    x=cow ; y=dog ; test -z $x ; echo $?
    a. **1**
    b. no output
    c. **0**
    d. **test: $x: integer expression expected**
    e. the number 0 or 1 followed by another 0 or 1 on a new line

103. What is the output of the following sequence of **bash** commands:
```
echo hi >wc ; wc wc >hi ; cat hi
```
a. **hi**
b. **0 0 0 wc**
c. no output
d. **1 1 2 wc**
e. **1 1 3 wc**

104. What is the **bash** shell output of this two-command sequence:
```
cd /home/alleni || echo "In $(pwd)"
```
a. **In $(pwd)**
b. **In /home/alleni**
c. no output
d. **"In $(pwd)"**
e. **In 0pwd)**

105. If **x=cow** and **y=dog** then what is the output of the following sequence of **bash** commands: **`[ $x = cow -a $y = cow ] ; echo $?`**
a. no output
b. **test: $x: integer expression expected**
c. **1**
d. the number 0 or 1 followed by another 0 or 1 on a new line
e. **0**

106. If **x=cow** and **y=dog** then what is the output of the following sequence of **bash** commands: **`[ $x = dog -o $y = cow ] ; echo $?`**
a. **0**
b. no output
c. **1**
d. **test: $x: integer expression expected**
e. the number 0 or 1 followed by another 0 or 1 on a new line

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

108. Given my directory **dir** and my file **dir/foo** owned by me, which permissions allow me to delete the file **dir/foo** from the directory, but not change the content (data) in the file?
a. Permissions **300** on directory **dir** and **500** on file **dir/foo**.
b. Permissions **300** on directory **dir** and **300** on file **dir/foo**.
c. Permissions **500** on directory **dir** and **400** on file **dir/foo**.
d. Permissions **100** on directory **dir** and **100** on file **dir/foo**.
e. Permissions **100** on directory **dir** and **200** on file **dir/foo**.

109. Which of these commands makes a file owned by me, also executable by me?
a. **umask 111 myfile**
b. **umask 777 myfile**
c. **chmod x+u myfile**
d. **chmod u+x ./myfile**
e. **chmod x=u ./myfile**

110. What is the output of this command sequence:
```
echo dog >one ; echo cow | head -2 one
```
a. an error message
b. **cow**
c. **dog** followed by **cow**
d. **cow** followed by **dog**
e. **dog**

Answer Key - DAT 2330 − Ian Allen − Fall 2003 - DAT 2330 Practice Test - 0%

Office use only: 44 61 84 99 60 56 28 77 40 98 14 21 109 90 24 89 20 87 79 67 108 29 64 110 16 82 9 17 32 45 8 37 88 30 83 12 103 104 15 107 51 65 13 94 52 75 35 69 78 71 53 18 73 97 38 3 41 4 23 7 48 39 43 1 5 47 102 54 46 74 36 57 66 92 76 100 68 95 42 49 62 70 96 63 6 31 26 101 85 2 58 106 55 25 93 80 34 33 22 59 50 27 19 81 11 10 105 72 91 86

1. e
2. c
3. e
4. b
5. d
6. d
7. a
8. d
9. d
10. b
11. d
12. c
13. c
14. b
15. d
16. e
17. d
18. b
19. e
20. c
21. c
22. d
23. d
24. a
25. e
26. a
27. b
28. e
29. b
30. c
31. d
32. a
33. d
34. a
35. a
36. a
37. a
38. e
39. b
40. e
41. b
42. e
43. c
44. b
45. b
46. e
47. b
48. c
49. b
50. b
51. e
52. a
53. e
54. e
55. e
56. b
57. a
58. b
59. d
60. c
61. a
62. e
63. b
64. b
65. c
66. b
67. a
68. a
69. b
70. c
71. b
72. d
73. d
74. c
75. b
76. e
77. a
78. b
79. a
80. c

81. e
82. a
83. c
84. c
85. e
86. d
87. c
88. c
89. b
90. a
91. d
92. a
93. d
94. b
95. e
96. d
97. a
98. a
99. b
100. d
101. d
102. a
103. e
104. c
105. c
106. c
107. b
108. a
109. d
110. e

Count of a: 22 20%
Count of b: 26 24%
Count of c: 20 18%
Count of d: 21 19%
Count of e: 21 19%

With 5 choices: 110
1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51
52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67
68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83
84 85 86 87 88 89 90 91
92 93 94 95 96 97 98 99
100 101 102 103 104 105
106 107 108 109 110

Macro .cmd splits: 52
Macro .ans splits: 0