

Evaluation: 158 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 - 158 Questions

This is a practice test containing many 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 will have approximately one question per minute. The real test may have some questions unlike anything given here.

(Office use only: 22 93 43 4 7 47 82 117 51 101 66 102 138 78 45 89 36 94 124 44 148 139 119 46 29 127 2 90 35 64 27 108 24 61 72 154 149 150 109 107 18 48 56 6 69 106 152 118 95 3 105 129 17 62 11 142 12 143 114 74 153 131 100 83 54 31 92 146 32 68 33 111 55 145 65 58 135 88 75 98 59 20 116 156 86 81 19 63 103 120 40 38 8 132 70 25 151 91 34 73 60 99 144 9 30 134 1 57 130 122 104 10 79 128 50 5 123 77 87 115 28 49 110 158 26 41 147 53 16 13 14 97 71 125 133 136 113 121 140 67 39 52 84 76 80 112 15 42 126 96 23 141 21 157 85 37 155 137)

1. 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 -eq :] ; then echo OK ; fi`
 - c. `if ["$foo" = *] ; then echo OK ; fi`
 - d. `if [$foo -eq ""] ; then echo OK ; fi`
 - e. `if ["$foo" = ""] ; then echo OK ; fi`
2. How many arguments and options are there to the command:


```
sort -r <infile
```

 - a. Two arguments, neither of which is an option.
 - b. One command line argument containing one option name.
 - c. A file name starting with a dash and an **<infile** switch option argument.
 - d. Two arguments, one of which is a single option name and the other is a pathname.
 - e. Three arguments, one of which contains an option and one is a pathname.
3. 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!=x] || echo foobar`
 - c. `[x = x] && echo foobar`
 - d. `[!x = y] && echo foobar`
 - e. `[x -ne y] || echo foobar`

4. 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 grep foo /etc/group ; then echo YES ; fi`
 - d. `if test foo = /etc/group ; then echo YES ; fi`
 - e. `if [grep foo /etc/group] ; then echo YES ; fi`
5. If **a=9** and **b=9**, which **bash** command sequence correctly compares the two numbers as equal and prints **OK**?
 - a. `if [$a -eq $b] ; then echo OK ; fi`
 - b. `if [$a==$b] ; then echo OK ; fi`
 - c. `if (a == b) ; then echo OK ; fi`
 - d. `if [a = b] ; then echo OK ; fi`
 - e. `if test a -eq b ; then echo OK ; fi`
6. What is the output of the following sequence of **bash** commands:


```
echo hi >wc ; wc wc >hi ; cat hi
```

 - a. `1 1 3 wc`
 - b. `0 0 0 wc`
 - c. no output
 - d. `1 1 2 wc`
 - e. `hi`
7. 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:


```
glob="*" ; echo "$glob"
```

 - a. `*`
 - b. `$glob`
 - c. the file names 1 through 123, surrounded by quotes
 - d. `"$glob"`
 - e. the file names 1 through 123
8. 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. `bar`
 - b. `new`
 - c. `$TERM`
 - d. `foo`
 - e. `TERM`

9. 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`
- 3
 - 1
 - 4
 - 2
 - 5
10. What is the output of this sequence of three shell commands:
`umask 762 ; touch newfile ; ls -l newfile`
- wx 1 me me 0 Oct 1 1:12 newfile
 - rw-rw--w- 1 me me 0 Oct 1 1:12 newfile
 - xr-x 1 me me 0 Oct 1 1:12 newfile
 - rwxrw--w- 1 me me 0 Oct 1 1:12 newfile
 - r-- 1 me me 0 Oct 1 1:12 newfile
11. What is the output of the following sequence of **bash** commands:
`a=cow ; b=dog ; touch $a ; test -z $a ; echo $?`
- the number 1 or 0 followed by another 1 or 0 on a new line
 - 1
 - `test: $a: integer expression expected`
 - no output
 - 0
12. What is the **bash** shell output of this two command sequence:
`PATH=/bin/ls:/bin/cat:/bin/sh ; cat nosuchfile`
- `ls: /bin/cat: command not found`
 - `bash: /bin/sh: No such file or directory`
 - `bash: cat: command not found`
 - `bash: /bin/ls: command not found`
 - `cat: nosuchfile: No such file or directory`
13. Given this long listing:
`drwxr-xr-x 448 me me 296 Dec 4 9:12 /dir`
 How many subdirectories lie immediately under `/dir`?
- 448
 - there is not enough information shown to answer the question
 - 446
 - 296
 - 294

14. What is in the file named **file** after this command sequence:
`echo a >c ; echo b >>c ; mv c d >file`
- no such file (nonexistent file)
 - a**
 - nothing - **file** is empty - no data
 - b**
 - a** followed by **b**
15. What is the output of the following sequence of **bash** commands:
`echo wc >wc ; wc wc >wc ; sort wc`
- `wc`
 - `1 1 2 wc`
 - `1 1 3 wc`
 - `0 0 0 wc`
 - no output
16. 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?
- `2 3 4`
 - `$2`
 - `"2 3 4"`
 - `"2`
 - `2`
17. Which **bash** command line below allows programs in the current directory to execute without preceding the names with `./`?
- `$PATH=/bin:./:$HOME`
 - `PATH = /bin:$HOME:.`
 - `PATH=/bin:$HOME:.`
 - `$PATH=.:$HOME:/bin`
 - `PATH = ./:$HOME:/bin`
18. How many arguments and options are there to the command:
`ls -lid /p`
- Two arguments, one of which is a single option name and the other is a pathname.
 - Two arguments: A file name starting with a dash and a `/p` switch option argument.
 - Two arguments, neither of which is an option.
 - Three arguments, one of which contains options and one is a pathname.
 - Two command line arguments, one of which contains three options.

19. 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,b,c`
 - `a=a b=b c=c`
 - `a;b;c`
 - `a b c`
 - `a:b:c`
20. What is the output of the following sequence of **bash** commands:
`echo wc >wc ; wc wc >wc ; cat wc`
- `0 0 0 wc`
 - no output
 - `1 1 3 wc`
 - `1 1 2 wc`
 - `wc`
21. Which of these first lines will cause this executable file to be interpreted using the Bash shell?
- `#!/bin/bash`
 - `!/bin/bash`
 - `/bin/bash -u`
 - `#!/bin/bash`
 - `!#/bin/bash -u`
22. What is the output of the following sequence of **bash** commands:
`wc='one two' ; test wc = wc`
- `0`
 - no output
 - `1 2 8 wc`
 - `test: too many arguments`
 - `1`
23. How can you ask the **bash** shell to complete commands or file names for you?
- You can type the first part of the command or file name and press the **TAB** key.
 - You can type the first part of the command or file name and press the **ALT** key.
 - Type **[CONTROL]-[D]** and the shell will present a menu of commands.
 - Type **[CONTROL]-[ALT]-[DEL]** and the shell will present a menu of commands.
 - Type **[ALT]-[F2]** the shell will present a menu of commands.

24. What is the output of the following sequence of **bash** commands:
`echo wc >wc ; wc wc >wc ; head wc`
- `1 1 2 wc`
 - `wc`
 - `1 1 3 wc`
 - `0 0 0 wc`
 - no output
25. If `a=cow` and `b=dog` then what is the output of the following sequence of **bash** commands: `[$a = cow -a $b = cow] ; echo $?`
- `test: $a: integer expression expected`
 - no output
 - `1`
 - `0`
 - the number 1 or 0 followed by another 1 or 0 on a new line
26. Which **bash** command sequence below always outputs just the date only if the first argument is either readable or executable?
- `if [-r -o -x "$1"]; then date ; fi`
 - `if ["-r $1" || "-x $1"]; then date ; fi`
 - `if ["$1" -eq -r -o "$1" -eq -x]; then date ; fi`
 - `if [-r || -x "$1"]; then date ; fi`
 - `if [-r "$1" -o -x "$1"]; then date ; fi`
27. Which **bash** command sequence correctly searches for the **chars** and then prints **OK** if it is found inside the password file?
- `if [grep chars /etc/passwd] ; then echo OK ; fi`
 - `if grep chars </etc/passwd ; then echo OK ; fi`
 - `if [test chars /etc/passwd] ; then echo OK ; fi`
 - `if test chars /etc/passwd ; then echo OK ; fi`
 - `if test chars = /etc/passwd ; then echo OK ; fi`
28. A shell script named **foo** is executed as follows:
`./foo 1 2 "3 4" 5`
Inside the script is the line: `echo "$3"`
What is the output from this line?
- `2 3 4`
 - `"3`
 - `1 2 3`
 - `$3`
 - `3 4`

29. 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**
- test: \$a: integer expression expected**
 - bash: cow: command not found**
 - no output
 - test: cow: integer expression expected**
 - cow**
30. 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?
- Permissions **300** on directory **dir** and **500** on file **dir/foo**.
 - Permissions **500** on directory **dir** and **400** on file **dir/foo**.
 - Permissions **100** on directory **dir** and **200** on file **dir/foo**.
 - Permissions **300** on directory **dir** and **300** on file **dir/foo**.
 - Permissions **100** on directory **dir** and **100** on file **dir/foo**.
31. If my current working directory is **/home**, and my home directory is **/home/xx**, which of the of the following commands copies the Unix password file into my home directory under the name **foo**?
- cp xx/../etc/passwd ../home/xx/foo**
 - cp ../../etc/passwd /xx/foo**
 - cp xx/../../etc/passwd xx/foo**
 - cp ../home/xx/../etc/passwd ./xx../foo**
 - cp ../etc/passwd ../xx/foo**
32. What is the link count of directory **dir** after this set of successful commands? **mkdir dir ; cd dir ; touch one ; mkdir two**
- 3
 - 1
 - 4
 - 5
 - 2
33. If variable **mt** might contain nothing (a null value - defined but empty), which **bash** command sequence correctly tests for this and prints **OK**?
- if [\$mt -eq ""] ; then echo OK ; fi**
 - if [\$mt -eq :] ; then echo OK ; fi**
 - if ["\$mt" = ""] ; then echo OK ; fi**
 - if ['\$mt' = ''] ; then echo OK ; fi**
 - if ["\$mt" = *] ; then echo OK ; fi**

34. 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'
```
- 3 characters
  - 13 characters
  - 1 character
  - 2 characters
  - 4 characters
35. What is the output of the following sequence of **bash** commands:
- ```
a=1 ; b=2 ; test $a -ge $b ; echo $?
```
- no output
 - the number 1 or 0 followed by another 1 or 0 on a new line
 - test: \$a: integer expression expected**
 - 0
 - 1
36. 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)?
- ddP**
 - ddp**
 - DDp**
 - :dp**
 - DDP**
37. Which command sequence below does *not* generate an error message from the last command in the sequence?
- mkdir one one/two ; rmdir one/two**
 - date >foo ; cp foo/. bar**
 - cat /etc/passwd > mail idallen@ncf.ca**
 - mkdir foo ; ln foo bar**
 - mkdir foo foo/bar ; rmdir foo**
38. Which line below is most likely to be the beginning of an error message?
- echo 1<&2 "... "**
 - echo 2>\$1 "... "**
 - echo 2<\$1 "... "**
 - echo 1>&2 "... "**
 - echo 2>&1 "... "**

39. What is the link count of directory **dir** after this set of successful commands?
- ```
mkdir dir ; cd dir ; touch foo ; mkdir a b c
```
- 4
  - 1
  - 3
  - 2
  - 5
40. 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?
- `./foo bar bar`
 - `./foo "$1" '$2'`
 - `./foo '$2' bar`
 - `./foo "bar" 'bar'`
 - `./foo $2 $2`
41. In an empty directory, what is the shell output of these three commands:
- ```
touch .1 .2 .3 11 12 ; b='.1* .2*' ; echo '$b'
```
- `$b`
  - `.1* .2*`
  - `' .1* .2* '`
  - `11 .1 12 .2`
  - `.1 .2`
42. Which of the following shell command lines displays all the names in the current directory that are exactly three letters (alphabetic) long (and nothing else)?
- `echo [a-zA,Z][a-zA,Z][a-zA,Z]`
  - `echo [a-mn-zA-YZ][ab-zA-B-YZ][za-yZA-Y]`
  - `echo [0-89][01-9][0-45-9]`
  - `echo ???`
  - `echo [azAZ][azAZ][azAZ]`
43. Which of these statements is true?
- If `/p` is an empty directory, `ls /p/*.*` produces an error message.
  - If `/q` is an empty directory, `echo /q/*.*` produces an error message.
  - Either single or double quotes will stop shell glob (wildcard) patterns from expanding.
  - The `ls dir` command looks up the directory argument **dir** in your `$PATH`.
  - Typing `./script` and `bash script` always give identical results.

44. Which **bash** command sequence correctly compares the two numbers and prints **OK**?
- `if [ 4 -gt 3 ] ; then echo OK ; fi`
  - `if [ ! 4 <= 3 ] ; then echo OK ; fi`
  - `if [ 4 > 3 ] ; then echo OK ; fi`
  - `if [ 4 -gr 3 ] ; then echo OK ; fi`
  - `if ( ! 4 < 3 ) ; then echo OK ; fi`
45. What is the output of the following sequence of **bash** commands:
- ```
x=0 ; y=1 ; touch $x ; test ! -z $x ; echo $?
```
- 1
 - `test: $x: integer expression expected`
 - 0
 - the number 1 or 0 followed by another 1 or 0 on a new line
 - no output
46. 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?
- `./foo $1 $1`
  - `./foo 1 "$1"`
  - `./foo bar '$1'`
  - `./foo bar 'bar'`
  - `./foo 'bar' "bar"`
47. Which command line displays the contents of the Unix **passwd** file one page at a time?
- `less </etc/passwd`
  - `/etc/passwd >less`
  - `cat /etc/passwd >less`
  - `less | /etc/passwd`
  - `/etc/passwd | less`
48. How can you ask the **bash** (Linux) shell to complete commands or file names for you?
- You can type the first part of the command or file name and press the **TAB** key.
  - Type `[CONTROL]-[ALT]-[DEL]` and the shell will present a menu of commands.
  - Type `[CONTROL]-[D]` and the shell will present a menu of commands.
  - Type `[ALT]-[F2]` the shell will present a menu of commands.
  - You can type the first part of the command or file name and press the **ALT** key.

49. How many arguments and options are there to the command:  
`cal -yj 1752`
- Two arguments, one of which is a single option and the other is a pathname.
  - Two arguments, neither of which is an option.
  - Three arguments, one of which contains options and one is a pathname.
  - Two command line arguments, one of which contains two options.
  - A single numeric option and a three-letter file name.
50. Which **bash** command sequence correctly searches for the **chars** and then prints **OK** if it is found inside the password file?
- `grep chars </etc/passwd && echo OK`
  - `grep chars >/etc/passwd && echo OK`
  - `grep chars >/etc/passwd || echo OK`
  - `grep chars </etc/passwd || echo OK`
  - `grep chars /etc/passwd || echo OK`
51. If `/bin/pig` is a program that outputs **hi** and `/usr/bin/pig` is a program that outputs **foo** what is the output of this shell command sequence: `PATH=/etc:/usr/bin:/bin ; pig`
- hi**
  - foo**
  - foo** followed by **hi**
  - hi** followed by **mom**
  - bash: pig: command not found**
52. What is the output of the following sequence of **bash** commands:
- ```

false && echo "linux rocks $?"

```
- `linux rocks 0`
 - no output
 - `linux rocks 1`
 - `linux rocks 0`
 - `linux rocks 1`
53. Which command sequence below outputs only lines 10-15 of the Unix password file?
- `head -10 /etc/passwd | tail -5 /etc/passwd`
 - `head -15 /etc/passwd | tail -5 /etc/passwd`
 - `tail -15 /etc/passwd | head -5`
 - `tail -10 /etc/passwd | head -15 /etc/passwd`
 - `head -15 /etc/passwd | tail -6`

54. 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'`
- 4 characters
 - 2 characters
 - 1 character
 - 13 characters
 - 3 characters
55. If `/etc/passwd` is a file name, which of the following pathnames always leads to the same file?
- `/etc/../../passwd`
 - `/etc/passwd/../../..`
 - `../etc/passwd`
 - `/etc/passwd/.`
 - `./etc/passwd`
56. 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`
- no lines (empty file)
 - 120**
 - 50**
 - 70**
 - 80**
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`
- 9**
 - 5**
 - 8**
 - 7**
 - 6**
58. 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`
- 40**
 - 60**
 - 200**
 - 160**
 - 100**

59. If **a=xxx** and **b=yyy** then what is the output of the following sequence of **bash** commands: **if \$a = \$b ; then echo \$a ; fi**
- test: xxx: integer expression expected**
 - test: \$a: integer expression expected**
 - bash: xxx: command not found**
 - xxx**
 - no output
60. What is the output of the following sequence of **bash** commands:
x=0 ; test \$x ; echo \$?
- test: \$x: integer expression expected**
 - no output
 - the number 0 or 1 followed by another 0 or 1 on a new line
 - 0
 - 1
61. Which Unix command sequence deletes a directory and everything inside it?
- deltree -all dir**
 - rm -all dir**
 - rmdir -r dir**
 - rm -r dir**
 - rmdir -all dir**
62. What is the **bash** shell output of this command sequence:
true && echo Hello There \$?
- Hello There ?**
 - no output
 - Hello There 1**
 - Hello There ?**
 - Hello There 0**
63. What is the output of this sequence of three shell commands:
umask 674 ; touch newfile ; ls -l newfile
- w--wxr-x 1 me me 0 Feb 20 07:55 newfile**
 - rw-rwxr-- 1 me me 0 Feb 20 07:55 newfile**
 - x----wx 1 me me 0 Feb 20 07:55 newfile**
 - w- 1 me me 0 Feb 20 07:55 newfile**
 - rw-rw-r-- 1 me me 0 Feb 20 07:55 newfile**
64. 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'**
- '\$cow'**
 - \$cow**
 - the file names 1 through 888
 - the file names 1 through 888, surrounded by quotes
 - *

65. What is the output of this command sequence:
echo pig >one ; echo cow | head -2 one
- pig**
 - cow**
 - an error message
 - pig** followed by **cow**
 - cow** followed by **pig**
66. If **x=cow** and **y=dog** then what is the output of the following sequence of **bash** commands: **[\$x = cow -a \$y = cow] ; echo \$?**
- no output
 - test: \$x: integer expression expected**
 - 1
 - 0
 - the number 0 or 1 followed by another 0 or 1 on a new line
67. How many arguments and options are there to the command:
wc -l <infile
- Two arguments, one of which is a single option name and the other is a pathname.
 - Three arguments, one of which contains an option and one is a pathname.
 - A file name starting with a dash and an **<infile** switch option argument.
 - One command line argument containing one option name.
 - Two arguments, neither of which is an option.
68. Which of the following **bash PATH** statements makes the most sense?
- PATH=/bin:/bin/cat:/usr/bin**
 - PATH=/bin:/usr/bin:/etc/passwd**
 - PATH=/bin/sh:/usr/bin:/etc:/bin**
 - PATH=/bin/ls:/etc:/usr/bin**
 - PATH=/bin:/usr/bin:/etc**
69. If **a=cow** and **b=pig** then what is the output of the following sequence of **bash** commands: **[\$a = pig -o \$b = pig] ; echo \$?**
- 0
 - the number 1 or 0 followed by another 1 or 0 on a new line
 - test: \$a: integer expression expected**
 - no output
 - 1

70. What is the output of the following sequence of **bash** commands:
`a=cow ; touch $a ; test -z $a ; echo $?`
- the number 1 or 0 followed by another 1 or 0 on a new line
 - no output
 - 1
 - `test: $a: integer expression expected`
 - 0
71. If `a=pig` and `b=dog` then what is the output of the following sequence of **bash** commands: `[$a = pig -a $b = pig] ; echo $?`
- the number 1 or 0 followed by another 1 or 0 on a new line
 - `test: $a: integer expression expected`
 - 0
 - 1
 - no output
72. Which of the following statements is true about this shell command line:
`>bar zoom bar haven`
- Error: The command name is missing from the command line.
 - The command `bar` sees three arguments.
 - The command `bar` sees only two arguments
 - The command `zoom` sees two arguments.
 - The command `zoom` sees three arguments.
73. Which of these statements is true?
- Typing `./script` and `bash script` always give identical results.
 - If `/y` is an empty directory, `echo /y/*` produces an error message.
 - If `/x` is an empty directory, `sort /x/*` produces an error message.
 - The `cat food` command looks up the file name argument `food` in your `$PATH`.
 - Only single quotes are strong enough to stop shell glob (wildcard) patterns from expanding.
74. 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?
- `rm dir/*1`
 - `rm dir/\??`
 - `rm dir/??`
 - `rm dir/1*`
 - `rm dir/?1`

75. Given my directory `dir` and my file `dir/bar` owned by me, which permissions allow me to delete the file `dir/bar` from the directory, but not change the content (data) in the file?
- Permissions `100` on directory `dir` and `200` on file `dir/bar`.
 - Permissions `300` on directory `dir` and `500` on file `dir/bar`.
 - Permissions `100` on directory `dir` and `100` on file `dir/bar`.
 - Permissions `500` on directory `dir` and `400` on file `dir/bar`.
 - Permissions `300` on directory `dir` and `300` on file `dir/bar`.
76. What is the **bash** shell output of this two-command sequence:
`cd /home/alleni || echo "In $(pwd)"`
- no output
 - `In /home/alleni`
 - `"In $(pwd)"`
 - `In 0pwd)`
 - `In $(pwd)`
77. How many arguments are passed to the command by the shell on this command line: `<pig pig -b "-a -r" >pig pig pig`
- 4
 - 6
 - 2
 - 3
 - 5
78. What is the output of the following sequence of **bash** commands:
`cd /bin && echo "cd $(pwd)"`
- `cd $(pwd)`
 - `bash: cd: /bin: No such file or directory`
 - no output
 - `cd 0pwd)`
 - `cd /bin`
79. 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?
- Permissions `600` on directory `dir` and `700` on file `dir/foo`.
 - Permissions `500` on directory `dir` and `600` on file `dir/foo`.
 - Permissions `100` on directory `dir` and `100` on file `dir/foo`.
 - Permissions `200` on directory `dir` and `200` on file `dir/foo`.
 - Permissions `400` on directory `dir` and `400` on file `dir/foo`.

80. What is the output of this sequence of three shell commands:
`umask 547 ; mkdir newdir ; ls -ld newdir`
- `d-w--wxrwx 1 me me 0 Feb 20 07:55 newdir`
 - `dr--r--rw- 1 me me 0 Feb 20 07:55 newdir`
 - `dr-xr--rwx 1 me me 0 Feb 20 07:55 newdir`
 - `d-w--w---- 1 me me 0 Feb 20 07:55 newdir`
 - `d-w--wx--- 1 me me 0 Feb 20 07:55 newdir`
81. What is the **bash** shell output of this two-command sequence:
`cd /etc/passwd && echo "in $(pwd)"`
- `in 0pwd)`
 - `bash: cd: /etc/passwd: Not a directory`
 - `in $(pwd)`
 - no output
 - `in /etc`
82. In an empty directory, what is the shell output of these three commands:
`touch xx .x xy .y xz ; a='x* y*' ; echo "$a"`
- `*x *y`
 - `xx xy`
 - `$a`
 - `x* y*`
 - `xx xy xz y*`
83. 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 $TERM`
- `linux`
 - `$TERM`
 - `foo`
 - `vt100`
 - `TERM`
84. What is the output of this sequence of three shell commands:
`echo x >abc ; ls >abc abc ; wc abc`
- no output
 - `1 1 4 abc`
 - `1 1 2 abc`
 - `1 1 3 abc`
 - `0 0 0 abc`

85. If **bar** is an executable script containing the line `cow=pig` then what is the **bash** output of this sequence of three commands:
`cow=cat ; ./bar ; echo "the '$cow' ate"`
- `the 'cat' ate`
 - `the $cow ate`
 - `the 'cow' ate`
 - `the 'pig' ate`
 - `the '$cow' ate`
86. 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 "$bat"`
- the file names 1 through 123, surrounded by quotes
 - `*`
 - `$bat`
 - the file names 1 through 123
 - `$bat`
87. In an empty directory, what is the shell output of these three commands:
`touch .1 .2 .3 11 12 ; a='.1* .2*' ; echo '$a'`
- `11 .1 12 .2`
 - `'.1* .2*'`
 - `.1 .2`
 - `$a`
 - `.1* .2*`
88. In an empty directory, what is the length of the longest file name created by the following **bash** shell two-command sequence:
`var='1 12 123 1234 12345' ; touch '$var'`
- 2 characters
 - 4 characters
 - 3 characters
 - 1 character
 - 13 characters
89. What is the **bash** shell output of this two command sequence:
`PATH=/bin/ls:/bin/head:/bin/sh ; head nosuchfile`
- `ls: /bin/head: command not found`
 - `bash: /bin/ls: command not found`
 - `bash: /bin/sh: No such file or directory`
 - `bash: head: command not found`
 - `head: nosuchfile: No such file or directory`

90. If **foo** were a file of text containing 50 different lines, what would be the output of this exact command line: **diff foo foo**
- an error message because **diff** doesn't allow the same file name twice
 - several lines, which are the lines that are different between the two files
 - an error message because **diff** only allows one file name
 - the contents of file **foo** would be displayed
 - no output
91. If **dog=12** and **cat=99** then which of the following **bash** command lines outputs only the word **hi** (and nothing else)?
- [dog -ne cat] || echo hi**
 - ![dog = cat] && echo hi**
 - [dog = dog] && echo hi**
 - [dog!=dog] || echo hi**
 - [dog -ne cat] && echo hi**
92. Which of the command lines below can generate a non-empty file?
- tail -5 /a/b >/a/b**
 - grep -v /a/b /a/b >/a/b**
 - tr abc ABC </a/b >/a/b**
 - sort -r /a/b >/a/b**
 - ls /a/b >/a/b**
93. If **x=5** and **y=5**, which **bash** command sequence correctly compares the two numbers as equal and prints **OK**?
- if test x -eq y ; then echo OK ; fi**
 - if (x == y) ; then echo OK ; fi**
 - if [x = y] ; then echo OK ; fi**
 - if test \$x -eq \$y ; then echo OK ; fi**
 - if [\$x==\$y] ; then echo OK ; fi**
94. What is the **bash** output of this command sequence:
- ```
false && echo "Hello There"
```
- "Hello There"**
  - no output
  - Hello There**
  - HelloThere**
  - Hello There**
95. What is the output of the following sequence of **bash** commands:
- ```
x=cow ; y=dog ; test -z $x ; echo $?
```
- no output
 - the number 0 or 1 followed by another 0 or 1 on a new line
 - test: \$x: integer expression expected**
 - 0
 - 1

96. If variable **bar** might contain nothing (a null value - defined but empty), which **bash** command sequence correctly tests for this and prints **YO**?
- if ['\$bar' = ''] ; then echo YO ; fi**
 - if [\$bar -eq :] ; then echo YO ; fi**
 - if [\$bar -eq ""] ; then echo YO ; fi**
 - if ["\$bar" = ""] ; then echo YO ; fi**
 - if ["\$bar" = *] ; then echo YO ; fi**
97. Which line below puts the count of the number of lines in the password file into the variable **foo**?
- foo=\$(cat -c /etc/passwd)**
 - foo=[wc /etc/passwd | echo \$1]**
 - foo=[cat -l /etc/passwd]**
 - foo=[grep -c /etc/passwd]**
 - foo=\$(wc -l </etc/passwd)**
98. 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?
- \$2**
  - b**
  - a bash error message: unbound (undefined) variable
  - b c d**
  - "b**
99. 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 $?
```
- no output
 - 0
 - 1
 - test: \$a: integer expression expected**
 - the number 1 or 0 followed by another 1 or 0 on a new line
100. What is the output of the following sequence of **bash** commands:
- ```
x=1 ; y=2 ; test $x -le $y ; echo $?
```
- 1
  - 0
  - test: \$x: integer expression expected**
  - no output
  - the number 0 or 1 followed by another 0 or 1 on a new line

101. What is the **bash** shell output of this two-command sequence:

```
cd /bin && echo "echo $(pwd)"
```

- `echo /bin`
- `echo 0pwd`
- `echo $(pwd)`
- no output
- `/bin`

102. What is the output of this sequence of three shell commands:

```
umask 457 ; mkdir newdir ; ls -ld newdir
```

- `d-wx-w---- 2 me me 512 Oct 1 1:12 newdir`
- `dr-xr-xrwx 2 me me 512 Oct 1 1:12 newdir`
- `dr--r-xrwx 2 me me 512 Oct 1 1:12 newdir`
- `d-wx-w-rwx 2 me me 512 Oct 1 1:12 newdir`
- `d-w--w---- 2 me me 512 Oct 1 1:12 newdir`

103. 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/????`

- no output
- `/dir/.123 /dir/.124`
- `echo: /dir/????: No such file or directory`
- `/dir/.123 /dir/.124 /dir/.???`
- `/dir/????`

104. If `cow=5` and `dog=5`, which **bash** command sequence correctly compares the two numbers as equal and prints **OK**?

- `if [ $cow==$dog ] ; then echo OK ; fi`
- `if test cow -eq dog ; then echo OK ; fi`
- `if test $cow -eq $dog ; then echo OK ; fi`
- `if [ cow = dog ] ; then echo OK ; fi`
- `if ( cow == dog ) ; then echo OK ; fi`

105. If `x=cow` and `y=dog` then what is the output of the following sequence of **bash** commands: `[ $x = dog -o $y = cow ] ; echo $?`

- the number 0 or 1 followed by another 0 or 1 on a new line
- no output
- `test: $x: integer expression expected`
- 1
- 0

106. How many arguments are passed to the command by the shell on this command line: `<bar bar -b "-a" '-r' >bar bar bar`

- 6
- 4
- 5
- 3
- 2

107. Which **bash** command sequence correctly searches for the **chars** and then prints **OK** if it is found inside the password file?

- `if test chars = /etc/passwd ; then echo OK ; fi`
- `if grep chars /etc/passwd ; then echo OK ; fi`
- `if [ grep chars /etc/passwd ] ; then echo OK ; fi`
- `if [ test chars /etc/passwd ] ; then echo OK ; fi`
- `if test chars /etc/passwd ; then echo OK ; fi`

108. What is the **bash** shell output of this two-command sequence:

```
cd /home/alleni && echo "In $(pwd)"
```

- `"In $(pwd)"`
- `In $(pwd)`
- no output
- `In /home/alleni`
- `In 0pwd)`

109. What is the **bash** shell output of this command sequence:

```
true && echo space junk $?
```

- no output
- `space junk ?`
- `space junk ?`
- `space junk 1`
- `space junk 0`

110. 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**?

- `11 22 33`
- `11:22:33`
- `xx=11 yy=22 zz=33`
- `11;22;33`
- `11,22,33`

111. 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`
- `hi`
  - `mom` followed by `hi`
  - `mom`
  - `bash: foo: command not found`
  - `hi` followed by `mom`
112. If `/etc/passwd` is a file name, which of the following pathnames always leads to the same file?
- `././etc/passwd`
  - `/etc/passwd/./.`
  - `/etc/./././passwd`
  - `./././etc/./passwd`
  - `/etc/passwd/././.`
113. Which command line below does not show any lines from inside the file `dog`?
- `more dog`
  - `head dog`
  - `ls dog`
  - `tail dog`
  - `less dog`
114. What is the output of the following sequence of `bash` commands:
- ```

false && echo "foo bar $?"

```
- `foo bar 1`
 - `foo bar 0`
 - `foo bar 0`
 - no output
 - `foo bar 1`
115. Which of the following shell command lines displays the names in the current directory that are exactly three numeric digits long (and nothing else)?
- `echo [1-31-31-3]`
 - `echo ???`
 - `echo [0-9][0-9][0-9]`
 - `echo [0-90-90-9]`
 - `echo [1-3][1-3][1-3]`

116. Which `bash` command sequence correctly compares the two numbers and prints `OK`?
- `if (! 4 < 3) ; then echo OK ; fi`
 - `if [4 > 3] ; then echo OK ; fi`
 - `if (let 4 > 3) ; then echo OK ; fi`
 - `if [! 4 <= 3] ; then echo OK ; fi`
 - `if [4 -gt 3] ; then echo OK ; fi`
117. Given the following `bash` shell command line:
- ```

read hi my ok

```
- which user keyboard input line below will assign the text `two` to the shell variable named `my`?
- `one,two,three`
  - `"one" "two" "three"`
  - `one two three`
  - `<one <two <three`
  - `hi=one my=two ok=three`
118. What is in the file named `file` after this command sequence:
- ```

echo a >x ; echo b >>x ; mv x y >file

```
- no such file (nonexistent file)
 - `b`
 - `a` followed by `b`
 - `a`
 - nothing - `file` is empty - no data
119. What is the output of the following sequence of `bash` commands:
- ```

cd /etc && echo "in $(pwd)"

```
- `bash: cd: /etc: No such file or directory`
  - `in /etc`
  - `in $(pwd)`
  - no output
  - `in 0pwd)`
120. 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"

```
- `$TERM`
 - `foo`
 - `vt100`
 - `TERM`
 - `linux`

121. If **a=cow** and **b=dog** then what is the output of the following sequence of **bash** commands: `[$a = dog -o $b = cow] ; echo $?`
- the number 1 or 0 followed by another 1 or 0 on a new line
 - 0
 - 1
 - no output
 - test: \$a: integer expression expected**
122. 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)?
- echo [azAZ][azAZ][azAZ]**
 - echo ???**
 - echo [a-zA-Z][a-zA-Z][a-zA-Z]**
 - echo [a-zA-Za-zA-Za-zA-Z]**
 - echo [a,zA,Z][a,zA,Z][a,zA,Z]**
123. Which of the following statements is true about this shell command line:
`>foo file bar haven`
- Error: The command name is missing from the command line.
 - The command **file** sees three arguments.
 - The command **foo** sees only two arguments
 - The command **file** sees two arguments.
 - The command **foo** sees three arguments.
124. Select the correct **bash** shell order of command line processing:
- aliases, variables, globs, redirection
 - aliases, globs, variables, redirection
 - redirection, aliases, globs, variables
 - aliases, variables, redirection, globs
 - aliases, redirection, variables, globs
125. 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**?
- cp ../home/me/../etc/passwd ./me../foo**
 - cp me/../../../../etc/passwd me/foo**
 - cp ../../etc/passwd /me/foo**
 - cp ./me../etc/passwd ../home/me/foo**
 - cp ../etc/passwd ../me/foo**

126. If **pig=12** and **cat=99** then which of the following **bash** command lines outputs only the word **hi** (and nothing else)?
- [pig -eq 12] || echo hi**
 - [pig = pig] && echo hi**
 - [pig -ne cat] && echo hi**
 - ![pig = cat] && echo hi**
 - [pig!=pig] || echo hi**
127. Which of these commands makes a file owned by me, also executable by me?
- chmod x+u myfile**
 - chmod u+x ./myfile**
 - chmod x=u ./myfile**
 - umask 777 myfile**
 - umask 111 myfile**
128. What is the output of this command sequence:
`echo dog >one ; echo cow | head -2 one`
- cow**
 - dog**
 - dog** followed by **cow**
 - cow** followed by **dog**
 - an error message
129. Which command sequence below outputs only lines 11-15 of the Unix password file?
- head -15 /etc/passwd | tail -5 /etc/passwd**
 - head -15 /etc/passwd | tail -5**
 - tail -15 /etc/passwd | head -10**
 - tail -10 /etc/passwd | head -15 /etc/passwd**
 - head -10 /etc/passwd | tail -15 /etc/passwd**
130. 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 -5 | head -1`
- 8
 - 7
 - 5
 - 9
 - 6

131. 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`
- 9
 - 8
 - 7
 - 5
 - 6
132. If the file **pig** contained the word **foo**, what would be the **bash** shell output of this two command sequence:
`PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/ls pig`
- foo
 - pig
 - no output
 - bash: /bin/ls: command not found**
 - /bin/ls: pig: No such file or directory**
133. What is the output of the following sequence of **bash** commands:
`x=1 ; touch x ; test ! -z $x ; echo $?`
- no output
 - the number 1 or 0 followed by another 1 or 0 on a new line
 - test: \$x: integer expression expected**
 - 0
 - 1
134. Which **bash** command sequence below always outputs just the date only if the first argument is either a file or a directory?
- `if [-f -o -d "$1"]; then date ; fi`
 - `if ["$1" -eq -f -o "$1" -eq -d]; then date ; fi`
 - `if [-f || -d "$1"]; then date ; fi`
 - `if ["-f $1" || "-d $1"]; then date ; fi`
 - `if [-f "$1" -o -d "$1"]; then date ; fi`
135. What is the **bash** shell output of this command sequence:
`false && echo "linux rocks $?"`
- no output
 - linux rocks 0
 - linux rocks 0
 - linux rocks 1
 - linux rocks 1

136. How many arguments are passed to the command by the shell on this command line: `<cow cow "-x" -y '-z' >cow cow`
- 2
 - 5
 - 4
 - 3
 - 6
137. If **x=pig** and **y=dog** then what is the output of the following sequence of **bash** commands: `if $x = $y ; then echo $y ; fi`
- no output
 - dog
 - test: \$x: integer expression expected**
 - bash: pig: command not found**
 - test: pig: integer expression expected**
138. If **happy** were a file of text containing 50 different lines, what would be the output of this exact command line: `diff happy happy`
- the contents of file **happy** would be displayed
 - no output
 - several lines, which are the lines that are different between the two files
 - an error message because **diff** doesn't allow the same file name twice
 - an error message because **diff** only allows one file name
139. In an empty directory, what is the **bash** shell output of this three-command sequence:
`touch aa .a ab .b .c ; x='.a* .b*' ; echo '$x'`
- `.a* .b*`
 - `$x`
 - `aa .a ab .b`
 - `.a .b`
 - `'.a* .b*'`
140. What is the output of the following sequence of **bash** commands:
`x=cow ; y=dog ; touch $x ; test -z $x ; echo $?`
- 0
 - 1
 - no output
 - test: \$x: integer expression expected**
 - the number 0 or 1 followed by another 0 or 1 on a new line
141. Which of the command lines below can generate a non-empty file?
- `sort -r /out >/out`
 - `tail -5 /out >/out`
 - `grep -v /out /out >/out`
 - `ls /out >/out`
 - `tr abc ABC </out >/out`

142. 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 ; ln z a`
- 4
 - 1
 - 3
 - 2
 - 5
143. 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 $foo`
- all the file names that start with an asterisk ('*')
 - `$foo`
 - the file names 1 through 765
 - *
 - an asterisk ('*') and the file names 1 through 765
144. Given my directory **dir** and my file **dir/bar** owned by me, which permissions allow me to change or create new content (data) in the file **dir/bar** but not delete the file?
- Permissions 600 on directory **dir** and 700 on file **dir/bar**.
 - Permissions 500 on directory **dir** and 600 on file **dir/bar**.
 - Permissions 200 on directory **dir** and 200 on file **dir/bar**.
 - Permissions 400 on directory **dir** and 400 on file **dir/bar**.
 - Permissions 100 on directory **dir** and 100 on file **dir/bar**.
145. Which command line below does not show any lines from inside the file **pig**?
- `more pig`
 - `head pig`
 - `less pig`
 - `ls pig`
 - `tail pig`
146. Which of the following shell command lines displays only the names in the current directory that are exactly three numeric digits long?
- `echo '[0-9]''[0-9]''[0-9]'`
 - `echo [0-9][0-9][0-9]`
 - `echo [1-3][1-3][1-3]`
 - `echo ???`
 - `echo '0-9'0-9'0-9'`

147. 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 -4 | tail -1`
- 7
 - 6
 - 5
 - 9
 - 8
148. If **a=1** and **b=2** then which of the following **bash** command lines outputs only the word **hi** (and nothing else)?
- `![a = b] && echo hi`
 - `[a!=a] || echo hi`
 - `[a -ne b] || echo hi`
 - `[a = a] && echo hi`
 - `[a -ne b] && echo hi`
149. Which **bash** command sequence below always outputs just the date only if the first argument is both not empty and a directory?
- `if ["$1" -eq -f -a "$1" -eq -d]; then date ; fi`
 - `if ["-s $1" && "-d $1"]; then date ; fi`
 - `if [-s && -d "$1"]; then date ; fi`
 - `if [-d "$1" -a -s "$1"]; then date ; fi`
 - `if [-s -a -d "$1"]; then date ; fi`
150. If the file **foo** contained the word **mom**, what would be the **bash** shell output of this two command sequence:
`PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/ls foo`
- `foo`
 - `/bin/ls: foo: No such file or directory`
 - `mom`
 - no output
 - `bash: /bin/ls: command not found`
151. If variable **cow** might contain nothing (a null value - defined but empty), which **bash** command sequence correctly tests for this and prints **OK**?
- `if ['$cow' = ''] ; then echo OK ; fi`
 - `if ["$cow" = *] ; then echo OK ; fi`
 - `if ["" = "$cow"] ; then echo OK ; fi`
 - `if [$cow -eq ""] ; then echo OK ; fi`
 - `if [$cow -eq :] ; then echo OK ; fi`

152. What is the output of the following sequence of **bash** commands:

```
date='October Monday' ; test date = date
```

- test: too many arguments**
 - 0
 - no output
 - 1
 - Mon Oct 27 17:01:38 EST 2003**
153. In an empty directory, what is the shell output of these three commands:
- ```
touch xx .x xy .y xz ; x='x* y*' ; echo "$x"
```
- \$x**
  - \*x \*y**
  - x\* y\***
  - xx xy xz y\***
  - xx xy**
154. What minimal permissions must you have on a directory to be able to execute successfully the command **ls .** from *inside* the directory?
- rw-**
  - x**
  - r-x**
  - r--**
  - wx**
155. If **bar** is an executable script containing the line **foo=dog** then what is the **bash** output of this sequence of three commands:
- ```
foo=cat ; ./bar ; echo "the '$foo' ate"
```
- the 'foo' ate**
 - the 'dog' ate**
 - the '\$foo' ate**
 - the \$foo ate**
 - the 'cat' ate**
156. Which **bash** command line below allows programs in the current directory to execute without preceding the names with **./**?
- PATH=./\$HOME:/usr/bin**
 - \$PATH=./\$HOME:/usr/bin**
 - PATH=/usr/bin:\$HOME:.**
 - PATH=/usr/bin/.\$HOME**
 - \$PATH=/usr/bin:.\$HOME**

157. 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
```

- sort "\$? \$? \$?"**
 - sort "\$1 \$2 \$3"**
 - sort "\$*"**
 - sort "\$@"**
 - sort "\$#"**
158. Given this long listing:
- ```
drwxr-xr-x 2048 bin bin 4096 Jan 2 14:22 /dir
```
- How many subdirectories lie immediately under **/dir**?
- there is not enough information shown to answer the question
  - 2048
  - 2046
  - 4094
  - 4096

**Answer Key - DAT 2330 – Ian Allen – Winter 2004 - DAT 2330  
Practice Test - 0%**

Office use only: 22 93 43 4 7 47 82 117 51 101 66 102 138 78 45 89 36 94 124 44 148 139 119 46 29 127 2 90 35 64 27 108 24 61 72 154 149  
150 109 107 18 48 56 6 69 106 152 118 95 3 105 129 17 62 11 142 12 143 114 74 153 131 100 83 54 31 92 146 32 68 33 111 55 145 65 58 135  
88 75 98 59 20 116 156 86 81 19 63 103 120 40 38 8 132 70 25 151 91 34 73 60 99 144 9 30 134 1 57 130 122 104 10 79 128 50 5 123 77 87 115  
28 49 110 158 26 41 147 53 16 13 14 97 71 125 133 136 113 121 140 67 39 52 84 76 80 112 15 42 126 96 23 141 21 157 85 37 155 137

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

- 81. b
  - 82. d
  - 83. a
  - 84. b
  - 85. a
  - 86. b
  - 87. d
  - 88. b
  - 89. d
  - 90. e
  - 91. c
  - 92. e
  - 93. d
  - 94. b
  - 95. e
  - 96. d
  - 97. e
  - 98. d
  - 99. b
  - 100. b
  - 101. a
  - 102. a
  - 103. e
  - 104. c
  - 105. d
  - 106. c
  - 107. b
  - 108. d
  - 109. e
  - 110. a
  - 111. c
  - 112. d
  - 113. c
  - 114. d
  - 115. c
  - 116. e
  - 117. c
  - 118. e
  - 119. b
  - 120. c
  - 121. c
  - 122. c
  - 123. d
  - 124. e
  - 125. b
  - 126. b
  - 127. b
  - 128. b
  - 129. b
  - 130. c
  - 131. c
  - 132. b
  - 133. d
  - 134. e
  - 135. a
  - 136. c
  - 137. d
  - 138. b
  - 139. b
  - 140. b
  - 141. d
  - 142. a
  - 143. c
  - 144. b
  - 145. d
  - 146. b
  - 147. e
  - 148. d
  - 149. d
  - 150. a
  - 151. c
  - 152. c
  - 153. c
  - 154. c
  - 155. e
  - 156. c
  - 157. d
  - 158. c
- Count of a: 29 18%  
 Count of b: 34 22%  
 Count of c: 38 24%  
 Count of d: 31 20%  
 Count of e: 26 16%
- With 5 choices: 158
- |    |    |    |    |    |    |    |    |   |    |    |
|----|----|----|----|----|----|----|----|---|----|----|
| 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  
106 107 108 109 110 111  
112 113 114 115 116 117  
118 119 120 121 122 123  
124 125 126 127 128 129  
130 131 132 133 134 135  
136 137 138 139 140 141  
142 143 144 145 146 147  
148 149 150 151 152 153  
154 155 156 157 158

Macro .cmd splits: 77  
Macro .ans splits: 0