Evaluation: 278 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 - 278 Questions

This is a practice test containing many practice questions. The real test will contain some questions similar to these. There are probably many more questions in this practice test than there will be time for in the real test. The real test will have approximately one question per minute.

The real test may have some questions unlike anything given here. Knowing the concepts behind the questions is necessary; memorizing these specific answers won't help.

(Office use only: 1 52 148 30 7 174 109 46 261 32 147 223 13 175 31 97 103 159 118 262 156 273 26 251 236 171 253 112 2 265 199 119 113 140 93 77 226 84 278 82 11 193 205 190 169 256 128 259 85 195 125 131 89 6 218 96 214 239 268 222 151 137 245 73 121 144 104 74 51 198 161 215 233 138 90 111 35 115 36 272 18 237 8 63 209 242 141 217 66 126 181 276 221 133 188 33 5 232 15 200 258 212 204 72 235 42 196 231 208 203 101 67 134 179 247 277 40 83 210 86 177 116 76 124 17 182 110 3 95 225 224 216 16 160 197 274 254 180 250 142 14 12 136 54 219 102 213 22 184 163 149 255 39 249 139 23 173 60 238 145 266 167 252 81 206 165 207 80 191 264 211 58 107 146 44 59 20 106 43 194 243 170 162 99 75 34 240 37 9 21 269 168 53 130 183 64 4 62 105 201 244 155 19 275 45 25 49 94 164 270 178 227 28 150 69 263 57 220 55 98 47 100 65 228 117 246 87 91 186 10 152 123 108 24 122 230 88 192 202 234 114 71 248 68 271 154 153 157 120 127 48 241 257 79 172 143 135 229 27 189 50 176 129 61 38 185 56 41 29 158 187 78 166 70 92 132 267 260)

1. A **`Makefile`** contains the following lines:
```
bar:
        rm one two
```
This means:
   a. if the user types "make two", items one and two will be removed
   b. if the user types "make one", items one and two will be removed
   c. the syntax "bar:" is not valid in a Makefile target
   d. if the user types "make rm", items one and two will be removed
   e. if the user types "make bar", items one and two will be removed

2. If **`happy`** were a file of text containing 50 different lines, what would be the output on your screen of this exact command line:
```
cp happy sad ; diff happy sad
```
   a. several lines, which are the lines that are different between the two files
   b. an error message because **`diff`** doesn't allow different file names
   c. the contents of file **`happy`** would be displayed
   d. no output
   e. an error message because **`diff`** only allows one file name

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

4. If **`/bin/foo`** is a program that outputs **`one`** and **`/usr/bin/foo`** is a program that outputs **`two`**, what is the output on your screen of this command sequence:
```
PATH=/dev:/usr/bin:/usr:/bin:/etc ; /bin/foo
```
   a. **`two`**
   b. **`one`**
   c. **`two`** followed by **`one`**
   d. **`bash: /bin/foo: command not found`**
   e. **`one`** followed by **`two`**

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

6. What is the output on your screen of this command sequence:
```
echo bat >pig ; echo one | tail pig
```
   a. **`bat`** followed by **`one`**
   b. **`one`** followed by **`bat`**
   c. an error message
   d. **`bat`**
   e. **`one`**

7. In an empty directory, what is the output on your screen 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*`**

8. If **`foo`** is a file containing the first column of the output of the **`last`** command, which command line shows the most frequent login?
   a. **`sort foo > uniq -c ; sort -nr uniq | head -1`**
   b. **`sort foo | uniq -c | sort -nr | head -1`**
   c. **`sort | uniq -c | sort -nr | head -1 foo`**
   d. **`uniq -c foo | sort -nr | head -1`**
   e. **`cat sort foo | uniq -c | sort -nr | head -1`**

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

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

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

12. Which command line below outputs only lines 10-15 of the 16-line file named **cow**?
a. **head -15 | tail -6 cow**
b. **head -16 cow | tail -5 cow**
c. **tail -16 cow | head -10**
d. **head -15 cow | tail -6**
e. **tail -10 cow | head -6 cow**

13. Given my directory **dir** and my file **dir/foo** owned by me, which permissions allow me to access and change the 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 **400** on directory **dir** and **400** on file **dir/foo**
c. Permissions **500** on directory **dir** and **600** on file **dir/foo**
d. Permissions **100** on directory **dir** and **100** on file **dir/foo**
e. Permissions **300** on directory **dir** and **200** on file **dir/foo**

14. What is the output on your screen of this command sequence:
```
echo pig >one ; echo bat | tail one
```
a. **bat** followed by **pig**
b. **bat**
c. an error message
d. **pig**
e. **pig** followed by **bat**

15. If **/bin/foo** is a program that outputs **one** and **/usr/bin/foo** is a program that outputs **two**, what is the output on your screen of this command sequence:
```
PATH=/etc:/usr/bin:/usr:/bin:/dev ; foo
```
a. **bash: foo: command not found**
b. **one** followed by **two**
c. **one**
d. **two**
e. **two** followed by **one**

16. In an empty directory, what appears on your screen after this command line?
```
ls out 2>/dev/null
```
a. no output
b. **ls: out: No such file or directory**
c. **out**
d. **ls: /dev/null: No such file or directory**
e. **ls: out 2>/dev/null: No such file or directory**

17. In an empty directory, what is the length of the longest file name after this sequence of commands?
```
echo hi >four ; cp four five5 ; mv five5 hi ; bzip2 hi
```
a. 3
b. 2
c. 5
d. 4
e. 6

18. What is the output on your screen of the following sequence of commands:
```
i=00 ; [ $i = 0 ] ; echo $?
```
a. **test: $i: integer expression expected**
b. no output
c. **0**
d. **1**
e. the number 0 or 1 followed by another 0 or 1 on a new line

19. What appears on your screen after this sequence of commands:
```
echo 1 >x ; ln x y ; echo 2 >>y ; head -1 x >y ; cat y
```
a. **2** followed by **1**
b. empty file - no output on the screen
c. **2**
d. **1**
e. **1** followed by **2**

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

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

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

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

24. Which correct command sequence below always outputs just the date only if the first argument is **both** not empty **and** a directory?
    a. **if ["$1" -eq -f -a "$1" -eq -d]; then date ; fi**
    b. **if ["-s $1" && "-d $1"]; then date ; fi**
    c. **if [-s && -d "$1"]; then date ; fi**
    d. **if [-s -a -d "$1"]; then date ; fi**
    e. **if [-d "$1" -a -s "$1"]; then date ; fi**

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

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

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

28. Select the correct **bash** shell order of command line processing:
    a. redirection, quotes, GLOBs, variables
    b. quotes, variables, redirection, GLOBs
    c. quotes, variables, GLOBs, redirection
    d. quotes, GLOBs, variables, redirection
    e. quotes, redirection, variables, GLOBs

29. A **Makefile** contains the following lines:
```
one:
        rm foo bar
```
    This means:
    a. if the user types "make one", items foo and bar will be removed
    b. if the user types "make bar", items foo and bar will be removed
    c. if the user types "make foo", items foo and bar will be removed
    d. if the user types "make rm", items foo and bar will be removed
    e. the syntax "one:" is not valid in a Makefile target

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

31. What is the output on your screen of this two-command sequence:
```
cd /bin && echo "echo $(pwd)"
```
    a. **echo /bin**
    b. **echo $(pwd)**
    c. **/bin**
    d. no output
    e. **echo 0pwd)**

32. What appears on your screen after this sequence of commands:
```
echo 1 >x ; ln x y ; echo 2 >>y ; sort x
```
    a. **1** followed by **2**
    b. **1**
    c. empty file - no output on the screen
    d. **2** followed by **1**
    e. **2**

33. The correct **g++** compiler suffix for a C++ source file is:
    a. **.cpp**
    b. **.C++**
    c. **.gpp**
    d. **.g++**
    e. **.cplus**

34. What is the output on your screen of the following command sequence:
```
cd /bin && echo "echo $(pwd)"
```
a. **echo /bin**
b. **echo 0pwd)**
c. **/bin**
d. **echo $(pwd)**
e. no output

35. In an empty directory, how many lines are in file **foo** after this command line:
```
ls  nosuchfile  .  ..  2>foo
```
a. **2**
b. **1**
c. **4**
d. empty file (no data)
e. **3**

36. If the file **bat** contained the word **foo**, what would be the output on your screen of this two command sequence:
```
PATH=/bin/cat:/bin/who:/bin/ls ; cat bat
```
a. **bash: cat: command not found**
b. **cat: bat: No such file or directory**
c. **foo**
d. **bat**
e. no output on screen

37. Which command line locates scripts in the **/bin** directory?
a. **cat /bin/* | file | grep script**
b. **file /bin | grep script**
c. **cat /bin | file | grep script**
d. **file /bin/* | grep script**
e. **ls /bin/* | file | grep script**

38. If variable **a** might contain nothing (a null value - defined but empty), which command sequence correctly tests for this and prints the date?
a. **if test "" = "$a" ; then date ; fi**
b. **if [$a = /dev/null] ; then date ; fi**
c. **if ["$a" = *] ; then date ; fi**
d. **if [’’’’ = ’’$a’’] ; then date ; fi**
e. **if test "" -eq $a ; then date ; fi**

39. what is the output on your screen of the following command sequence:
```
true && echo Linux     Rocks $?
```
a. **Linux Rocks $?**
b. no output
c. **Linux Rocks 0**
d. **Linux Rocks 1**
e. **Linux Rocks ?**

40. If the file **pig** contained the word **bar**, what would be the output on your screen of this two command sequence:
```
PATH=/etc/passwd:/bin/ls:/bin/who ; /bin/cat pig
```
a. **bar**
b. **bash: /bin/cat: command not found**
c. no output on screen
d. **/bin/cat: pig: No such file or directory**
e. **pig**

41. 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?
a. Permissions **500** on directory **dir** and **400** on file **dir/bar**.
b. Permissions **300** on directory **dir** and **300** on file **dir/bar**.
c. Permissions **100** on directory **dir** and **100** on file **dir/bar**.
d. Permissions **300** on directory **dir** and **500** on file **dir/bar**.
e. Permissions **100** on directory **dir** and **200** on file **dir/bar**.

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

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

44. What is the output on your screen of this sequence of three shell commands:
```
umask 762 ; touch newfile ; ls -l newfile
```
a. **-rwxrw--w- 1 me me 0 Oct 1 1:12 newfile**
b. **------xr-x 1 me me 0 Oct 1 1:12 newfile**
c. **-------r-- 1 me me 0 Oct 1 1:12 newfile**
d. **-rw-rw--w- 1 me me 0 Oct 1 1:12 newfile**
e. **--------wx 1 me me 0 Oct 1 1:12 newfile**

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

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

47. What is the link count of directory **d** after this set of successful commands?
    mkdir d ; cd d ; touch f ; ln f x ; ln f y
    a. 5
    b. 4
    c. 2
    d. 3
    e. 1

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

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

50. What is the output on your screen 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 "$bat"**
    a. the file names **1** through **123**
    b. *****
    c. the file names **1** through **123**, surrounded by quotes
    d. **$bat**
    e. **"$bat"**

51. What is the correct syntax to redirect both standard output and standard error into the same output file?
    a. **sum 2>1 >out foo**
    b. **sum 1>out 2>out foo**
    c. **sum >out foo 2>&1**
    d. **sum foo 1>out 2>1**
    e. **sum 2>&1 foo >out**

52. What is the link count of directory **x** after this set of successful commands?
    mkdir x ; mkdir x/y ; mkdir x/z ; mkdir x/y/z
    a. 2
    b. 4
    c. 3
    d. 5
    e. 1

53. If your PATH variable contains **/bin:/usr/bin**, what is the output on your screen of this command line? **echo '$PATH'**
    a. **echo: $PATH: No such file or directory**
    b. **$PATH**
    c. **'/bin:/usr/bin'**
    d. **'$PATH'**
    e. **/bin:/usr/bin**

54. 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 on your screen from this line?
    a. **$3**
    b. **1 2 3**
    c. **2 3 4**
    d. **3 4**
    e. **"3**

55. Which 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. **$PATH=.:$HOME:/bin**
    e. **PATH=/bin:$HOME:.**

56. In an empty directory, what appears on your screen after this command line?
    ls 2>/dev/null nosuchfile
    a. no output
    b. **ls: nosuchfile: No such file or directory**
    c. **ls: /dev/null: No such file or directory**
    d. **nosuchfile**
    e. **ls: 2>/dev/null nosuchfile: No such file or directory**

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

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

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

60. Which command line below allows the shell to execute programs in the current directory without preceding the names with **`./`**?
    a. **`PATH=/etc:/dev:.`**
    b. **`PATH=/usr/bin/.:/dev/null`**
    c. **`PATH=./dev:./bin`**
    d. **`PATH=./dev:/usr/bin`**
    e. **`PATH=/bin/.:..`**

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

62. What is the output on your screen of the following command sequence if run in a directory containing 123 files with names that are all the numbers from **`1`** to **`123`** inclusive: **`glob="*" ; echo "$glob"`**
    a. the file names **`1`** through **`123`**
    b. **`*`**
    c. **`"$glob"`**
    d. **`$glob`**
    e. the file names **`1`** through **`123`**, surrounded by quotes

63. Which 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`**

64. If directory **`/000`** contains these three four-character file names: **`.abc`**, **`.xyz`**, **`.???`**, then what is the output on your screen of the following command line:
    `echo /000/????`
    a. **`/000/????`**
    b. **`echo: /000/????: No such file or directory`**
    c. **`/000/.abc /000/.xyz`**
    d. no output
    e. **`/000/.abc /000/.xyz /000/.???`**

65. What is in the file **`cow`** after this command line:
    `echo a >b ; echo b >a ; mv b a >cow`
    a. **`a`** followed by **`b`**
    b. no such file (nonexistent)
    c. **`b`**
    d. **`a`**
    e. nothing - empty file - no data

66. What is the output on your screen of the following command sequence:
    `echo wc >wc ; wc wc >wc ; head wc`
    a. no output
    b. **`wc`**
    c. **`1 1 2 wc`**
    d. **`0 0 0 wc`**
    e. **`1 1 3 wc`**

67. In an empty directory, what is the length of the longest file name after this sequence of commands?
    `echo hi >sixsix ; mv sixsix four ; cp four hi ; gzip hi`
    a. 4
    b. 6
    c. 2
    d. 3
    e. 5

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

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

70. What is the output on your screen of this two-command sequence:
```
cd /bin && echo "cd $(pwd)"
```
a. no output
b. **cd $(pwd)**
c. **cd /bin**
d. **cd 0pwd)**
e. **/bin**

71. What is the output on your screen of the following sequence of commands:
```
x=0 ; [ $x = 00 ] ; echo $?
```
a. **0**
b. no output
c. **1**
d. the number 0 or 1 followed by another 0 or 1 on a new line
e. **test: $x: integer expression expected**

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

73. Which command sequence below always outputs just the date only if the first argument is both a directory and not empty?
a. **if [-s -a -d "$1"]; then date ; fi**
b. **if ["$1" -eq -f -a "$1" -eq -d]; then date ; fi**
c. **if [-n "$1" -o -d "$1"]; then date ; fi**
d. **if [-s "$1" -a -d "$1"]; then date ; fi**
e. **if ["-s $1" && "-d $1"]; then date ; fi**

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

75. In a directory containing one file named **dog**, what appears on your screen after this command line? **1>/dev/null ls ***
a. no output on screen
b. **dog**
c. **bash: 1>/dev/null: command not found**
d. *****
e. **ls: *: No such file or directory**

76. In response to the following command line: **read var1 var2 var3** which user keyboard input line below will assign the text **three** to the shell variable named **var3**?
a. **one,two,three**
b. **one:two:three**
c. **var1=one var2=two var3=three**
d. **one two three**
e. **$var1="one" $var2="two" $var3="three"**

77. If **bar** is a script containing the line **TERM=vt100 ; export TERM** what is the output on your screen of the following command sequence:
```
TERM=linux ; ./bar ; echo $TERM
```
a. **TERM**
b. **bar**
c. **linux**
d. **vt100**
e. **$TERM**

78. The default output file generated by the C and C++ compilers is named:
a. **a.out**
b. **argv**
c. **a.cpp**
d. **a.c++**
e. **a.o**

79. If **bar** is an executable script containing the line **dog=bat** then what is the output on your screen of this sequence of three commands:
```
dog=cat ; ./bar ; echo "the ’$dog’ ate"
```
a. **the ’bat’ ate**
b. **the ’$dog’ ate**
c. **the $dog ate**
d. **the ’dog’ ate**
e. **the ’cat’ ate**

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

81. How can you ask the **bash** (Linux) shell to complete commands or file names for you?
a. Type the first part of the command or file name and press the **[ALT]-[F1]** key.
b. Type the first part of the command or file name and press the **[TAB]** key.
c. Type the first part of the command or file name and press the **[CTRL]-[D]** key.
d. Type the first part of the command or file name and press the **[ALT]** key.
e. Type the first part of the command or file name and press the **[CTRL]-[C]** key.

82. Which command sequence below does *not* generate an error message from the last command in the sequence?
    a. **`mkdir foo ; touch foo/bar ; rmdir foo`**
    b. **`mkdir foo bar ; ln foo xxx`**
    c. **`mkdir ddd ddd/fff fff ; rmdir ddd/fff`**
    d. **`cat /etc/passwd > mail idallen@idallen.ca`**
    e. **`date >foo ; cp foo/. bar`**

83. Given my directory **`dir`** and my file **`dir/bar`** owned by me, which permissions allow me to access and change the content (data) in the file **`dir/bar`** but not delete the file?
    a. Permissions **`300`** on directory **`dir`** and **`500`** on file **`dir/bar`**.
    b. Permissions **`700`** on directory **`dir`** and **`300`** on file **`dir/bar`**.
    c. Permissions **`500`** on directory **`dir`** and **`500`** on file **`dir/bar`**.
    d. Permissions **`600`** on directory **`dir`** and **`200`** on file **`dir/bar`**.
    e. Permissions **`100`** on directory **`dir`** and **`200`** on file **`dir/bar`**.

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

85. What is true about this output from **`ls -il foo bar`**
```
35 -rw-rw-r-- 2 bin bin  3 Jan 24 01:03 foo
36 -rw-rw-r-- 2 bin bin  3 Jan 24 01:03 bar
```
    a. **`foo`** and **`bar`** are two of three names for this file
    b. this output is not possible
    c. **`foo`** and **`bar`** each have two names (four names total)
    d. **`foo`** and **`bar`** each have three names (six names total)
    e. **`foo`** and **`bar`** are names for the same file

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

87. What is the output on your screen of the following command sequence:
```
cd /etc/passwd && echo "in $(pwd)"
```
    a. **`in 0pwd)`**
    b. **`in /etc`**
    c. **`bash: cd: /etc/passwd: Not a directory`**
    d. **`in $(pwd)`**
    e. no output

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

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

90. What is the correct syntax to redirect both standard output and standard error into the same output file?
    a. **`wc 2>1 >out foo`**
    b. **`wc >out 2>1 foo`**
    c. **`wc >out 2>&1 foo`**
    d. **`wc >out 2>out foo`**
    e. **`wc 2>&1 >out foo`**

91. What is the output on your screen of this sequence of three shell commands:
```
echo ls >fil ; >fil ls fil ; wc fil
```
    a. **`1 1 4 fil`**
    b. **`0 0 0 fil`**
    c. **`1 1 2 fil`**
    d. no output
    e. **`1 1 3 fil`**

92. Which of these statements is true?
    a. You can only login to Unix once per userid; you cannot be logged in to the same machine twice.
    b. To indicate End-of-File (no more input), type [CTRL]-[C].
    c. To erase an entire line of typing, type [CTRL]-[D].
    d. Unix commands can be entered in upper-case or lower-case letters; they are equivalent.
    e. Unix commands must be entered in lower-case letters.

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

94. 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
```
a. 2
b. 3
c. 4
d. 1
e. 5

95. What is the output on your screen of this sequence of three shell commands:
```
umask 574 ; mkdir newdir ; ls -ld newdir
```
a. `d-w-----w- 1 me me 0 Oct 1 07:55 newdir`
b. `dr-xrwxr-- 1 me me 0 Oct 1 07:55 newdir`
c. `dr--rw-r-- 1 me me 0 Oct 1 07:55 newdir`
d. `d-w-rwx-wx 1 me me 0 Oct 1 07:55 newdir`
e. `d-w-----wx 1 me me 0 Oct 1 07:55 newdir`

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

97. A shell script named **bar** is executed as follows:
```
./bar "a b" "c d e" f
```
Inside the script is the line: **echo "$2"**
What is the output on your screen from this line?
a. **b**
b. **c d e**
c. **a b**
d. **b"**
e. **$2**

98. Which command sends a file to a remote machine **foo.ca**?
a. **cp one foo.ca:two**
b. **cat one >foo.ca:two**
c. **scp one >foo.ca:two**
d. **mv one foo.ca:two**
e. **scp one foo.ca:two**

99. 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 **300** on file **dir/foo**.
b. Permissions **100** on directory **dir** and **100** on file **dir/foo**.
c. Permissions **100** on directory **dir** and **200** on file **dir/foo**.
d. Permissions **500** on directory **dir** and **400** on file **dir/foo**.
e. Permissions **300** on directory **dir** and **500** on file **dir/foo**.

100. What is the output on your screen of this two-command sequence:
```
cd /bin/ls && echo "in $(pwd)"
```
a. **in $(pwd)**
b. **in 0pwd)**
c. **in /etc**
d. **bash: cd: /bin/ls: Not a directory**
e. no output

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

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

103. What is the possible output on your screen of this command line:
```
echo wc >date ; sort date >date ; cat date
```
a. **wc**
b. **Mon Sep 27 15:58:34 EDT 2004**
c. **1 6 28 date**
d. no output on screen
e. **1 6 29 date**

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

105. Which 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 [ "-f $1" || "-d $1" ]; then date ; fi`
    c. `if [ "$1" -eq -f -o "$1" -eq -d ]; then date ; fi`
    d. `if [ -f -o -d "$1" ]; then date ; fi`
    e. `if [ -f "$1" -o -d "$1" ]; then date ; fi`

106. If **dog** is an executable script containing the line: `umask 0002`
what is the output on your screen of the following sequence of commands:
`umask 0077 ; ./dog ; umask`
    a. no output on screen
    b. **0079**
    c. **0077**
    d. **0002**
    e. **0075**

107. What is the output on your screen 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 $foo`
    a. all the file names that start with an asterisk ('*****')
    b. an asterisk ('*****') and the file names **1** through **765**
    c. **$foo**
    d. the file names **1** through **765**
    e. *****

108. Which command line tells you the recursive count of all pathnames under the current directory and all subdirectories?
    a. `wc "$PATH"`
    b. `wc .`
    c. `find | wc`
    d. `ls | wc`
    e. `wc *`

109. What is true about this output from `ls -il foo bar`
```
23 -rwxr----- 3 root root 2 Jul 31 12:33 foo
24 -rwxr----- 3 root root 2 Jul 31 12:33 bar
```
    a. this output is not possible
    b. **foo** and **bar** are names for the same file
    c. **foo** and **bar** are two of three names for this file
    d. **foo** and **bar** are names for different files
    e. **foo** and **bar** each have two names (four names total)

110. What is the output on your screen of this two-command sequence:
`cd /home || echo "cd $(pwd)"`
    a. `cd $(pwd)`
    b. `cd /home`
    c. `cd 0pwd)`
    d. no output
    e. **/home**

111. In an empty directory, what is the length of the longest file name (including extension) after this sequence of commands?
`date >four ; cp four five5 ; mv five5 hi ; bzip2 hi`
    a. 3
    b. 5
    c. 6
    d. 7
    e. 4

112. If **x=one** and **y=two** then what is the output on your screen of the following sequence of commands: `if $x = $y ; then echo $a ; fi`
    a. **one**
    b. no output
    c. **test: $x: integer expression expected**
    d. **bash: one: command not found**
    e. **test: one: integer expression expected**

113. What is the output on your screen if a user signals an end-of-file from the keyboard during this command sequence? `read input && echo $?`
    a. no output on screen
    b. **0**
    c. an error message
    d. **$?**
    e. **1**

114. What is the output on your screen of this command sequence:
`true && echo Linux Rocks $?`
    a. **Linux Rocks ?**
    b. **Linux Rocks ?**
    c. **Linux Rocks 0**
    d. no output
    e. **Linux Rocks 1**

115. Which 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 [ grep chars /etc/passwd ] ; then echo OK ; fi`
    d. `if [ test chars /etc/passwd ] ; then echo OK ; fi`
    e. `if test chars = /etc/passwd ; then echo OK ; fi`

116. what is the output on your screen of the following command sequence:
`false && echo "hello there $?"`
    a. no output
    b. **hello there 1**
    c. **hello there 0**
    d. **hello there 0**
    e. **hello there 1**

117. If **cat** is an executable script containing the line:
```
TERM=linux ; export TERM
```
what is the output on your screen of the following sequence of commands:
```
TERM=vt100 ; ./cat ; echo "$TERM"
```
a. **vt100**
b. **$TERM**
c. **linux**
d. **TERM**
e. **cat**

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

119. What is true about this output from **ls -il foo bar**
```
72 -rwxrwxrwx 2 bin bin  3 Oct 30 09:23 foo
72 -r--r--r-- 2 bin bin  3 Oct 30 09:23 bar
```
a. **foo** and **bar** are names for the same file
b. this output is not possible
c. **foo** and **bar** are two of three names for this file
d. **foo** and **bar** each have two names (four names total)
e. **foo** and **bar** are names for different files

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

121. What is the output on your screen of this command sequence:
```
false && echo "linux     rules $?"
```
a. **linux rules 0**
b. **linux rules 1**
c. **linux rules 0**
d. **linux rules 1**
e. no output

122. What appears on your screen after this command line?
```
echo hi >ls ; cat ls > wc
```
a. **1 1 2**
b. no output on screen
c. **1 1 3**
d. **ls**
e. **hi**

123. If file **foo** occupies two disk blocks, how many disk blocks are in use after this sequence of commands:
```
cp foo bar ; ln bar one ; cp one two ; ln one ten
```
a. 4 blocks
b. 8 blocks
c. 10 blocks
d. 6 blocks
e. 2 blocks

124. What is the correct syntax to redirect both standard output and standard error into the same output file?
a. **ls -l 2>$1 >foo**
b. **ls -l >foo 2>&1**
c. **ls -l 2>&1 >foo**
d. **ls -l >foo 2>$1**
e. **ls -l >foo 2>foo**

125. Given the following 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. **xx=11 yy=22 zz=33**
c. **11:22:33**
d. **11,22,33**
e. **11 22 33**

126. What is the output on your screen 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 2 abc**
e. **1 1 3 abc**

127. In an empty directory, what is the output on your screen of this three-command sequence: **touch za .z zb .w .1 ; f=".z* .w*" ; echo '$f'**
a. **.z* .w***
b. **za .z zb .w**
c. **.z .w**
d. **za zb .w***
e. **$f**

128. A **Makefile** contains the following target: **ant: foo bar**
which means:
a. item bar depends on items ant and foo
b. items ant and foo depend on item bar
c. item ant depends on items foo and bar
d. items foo and bar depend on item ant
e. the syntax "ant:" is not valid in a Makefile target

129. In an empty directory, what appears on your screen after this command line?
ls 1>/dev/null nosuchfile
a. no output
b. **ls: /dev/null: No such file or directory**
c. **ls: 1>/dev/null nosuchfile: No such file or directory**
d. **ls: nosuchfile: No such file or directory**
e. **nosuchfile**

130. Which command line copies all the files from directory **a** to directory **b**?
a. **cd a ; tar -r /tmp/i . ; cd ../b ; tar -rvx /tmp/i**
b. **cd a ; tar czf /tmp/i . ; cd ../b ; tar xvf /tmp/i**
c. **cd a ; tar czf /tmp/i . ; cd ../b ; tar xzf /tmp/i**
d. **cd a ; tar xf /tmp/i . ; cd ../b ; tar czvf /tmp/i**
e. **cd a ; tar -rc /tmp/i . ; cd ../b ; tar -rx /tmp/i**

131. Which command line copies all the files from directory **a** to directory **b**?
a. **cd a ; tar cf /tmp/i . ; cd ../b ; tar xf /tmp/i**
b. **cd a ; tar xvf /tmp/i . ; cd ../b ; tar czf /tmp/i**
c. **cd a ; tar czf /tmp/i . ; cd ../b ; tar xvf /tmp/i**
d. **cd a ; tar -r /tmp/i . ; cd ../b ; tar -rx /tmp/i**
e. **cd a ; tar -rc /tmp/i . ; cd ../b ; tar -rx /tmp/i**

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

133. Given the following command line: **read one two three**
which user keyboard input line below will assign the text **bb** to the shell variable named **two**?
a. **aa bb cc**
b. **aa;bb;cc**
c. **aa,bb,cc**
d. **one=aa two=bb three=cc**
e. **aa:bb:cc**

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

135. What is the output on your screen 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 '$cow'**
a. the file names **1** through **888**, surrounded by quotes
b. **$cow**
c. *
d. the file names **1** through **888**
e. **'$cow'**

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

137. 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 "$1 $2 $3"**
b. **sort "$@"**
c. **sort "$? $? $?"**
d. **sort "$#"**
e. **sort "$*"**

138. What is the output on your screen of this sequence of three shell commands:
echo ls >cat ; >cat ls cat ; wc cat
a. **1 1 3 cat**
b. no output
c. **1 1 2 cat**
d. **1 1 4 cat**
e. **0 0 0 cat**

139. Which command tells you the full absolute pathname of the **lynx** command?
a. **absolute lynx**
b. **whereis lynx**
c. **whereis | grep lynx**
d. **echo "$PATH" | grep lynx**
e. **absolute "$PATH" | grep lynx**

140. What is the output on your screen of the following command sequence:
```
date='Friday March 12' ; test date = date
```
a. **Fri Mar 12 10:20:39 EST 2004**
b. **0**
c. no output
d. **test: too many arguments**
e. **1**

141. 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 **100** on directory **dir** and **100** on file **dir/foo**
b. Permissions **400** on directory **dir** and **400** on file **dir/foo**
c. Permissions **500** on directory **dir** and **600** on file **dir/foo**
d. Permissions **200** on directory **dir** and **200** on file **dir/foo**
e. Permissions **600** on directory **dir** and **700** on file **dir/foo**

142. Given my directory **dir** and my file **dir/fil** owned by me, which permissions allow me to access and change the content (data) in the file but not delete the file?
a. Permissions **500** on directory **dir** and **500** on file **dir/fil**
b. Permissions **600** on directory **dir** and **200** on file **dir/fil**
c. Permissions **300** on directory **dir** and **500** on file **dir/fil**
d. Permissions **100** on directory **dir** and **200** on file **dir/fil**
e. Permissions **700** on directory **dir** and **300** on file **dir/fil**

143. What is the output on your screen if a user signals an end-of-file from the keyboard during this command sequence? **read input || echo $?**
a. an error message
b. no output on screen
c. **$?**
d. **1**
e. **0**

144. If **one** were a file of text containing 10 different lines, what would be the output on your screen of this command line: **cp one two ; diff one two**
a. no output on screen
b. an error message because **diff** only allows one file name
c. an error message because **diff** doesn't allow different file names
d. the contents of one of the files would be displayed
e. several lines, which are the lines that are different between the two files

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

146. In an empty directory, what is the length of the longest file name (including extension) after this sequence of commands?
```
date >sixsix ; cp sixsix no ; mv sixsix four ; gzip no
```
a. 3
b. 2
c. 6
d. 5
e. 4

147. What will appear on your screen if you execute this sequence of commands:
```
echo 1 >a ; ln a b ; echo 2 >b ; chmod 266 b ; cat a
```
a. an error message
b. **1**
c. no output on screen
d. **2**
e. **1** followed by **2**

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

149. What is the output on your screen of this sequence of three shell commands:
```
umask 457 ; mkdir dir ; ls -ld dir
```
a. **d-wx-w---- 2 me me 128 Jan 9 9:34 dir**
b. **d-w--w---- 2 me me 128 Jan 9 9:34 dir**
c. **dr--r-xrwx 2 me me 128 Jan 9 9:34 dir**
d. **dr-xr-xrwx 2 me me 128 Jan 9 9:34 dir**
e. **d-wx-w-rwx 2 me me 128 Jan 9 9:34 dir**

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

151. What is the output on your screen of the following sequence of commands:
```
a=9 ; b=9 ; [ $a -le $b ] ; echo $?
```
a. the number 1 or 0 followed by another 1 or 0 on a new line
b. **test: $a: integer expression expected**
c. **1**
d. no output
e. **0**

152. Which line below puts the count of the number of lines in the password file into the variable **foo**?
   a. **foo=$(cat -c /etc/passwd)**
   b. **foo=$(wc /etc/passwd | awk echo $1)**
   c. **foo=$(wc -l /etc/passwd | awk "print $1")**
   d. **foo=$(awk -F: /etc/passwd | wc -l)**
   e. **foo=$(wc -l </etc/passwd)**

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

154. Which command tells you the count of lines in the **bash** manual page?
   a. **man bash > wc ; cat wc**
   b. **whereis bash | wc**
   c. **man bash | wc**
   d. **apropos bash | wc**
   e. **which bash | wc**

155. What is the output on your screen of the following command sequence:
   a=sky ; touch $a ; test -z $a ; echo $?
   a. **sky**
   b. **1**
   c. **0**
   d. **test: $a: integer expression expected**
   e. no output

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

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

158. If **a=cow** and **b=dog** then what is the output on your screen of the following sequence of commands: **[$a = cow -a $b = cow] ; 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

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

160. What is the output on your screen of the following command sequence:
   f=1 ; touch f ; test ! -z $f ; echo $?
   a. **0**
   b. **test: $f: integer expression expected**
   c. **1**
   d. the number 1 or 0 followed by another 1 or 0 on a new line
   e. no output

161. Which of the following shell command lines displays the names in the current directory that are exactly three numeric digits long (and nothing else)?
   a. **echo [1-3][1-3][1-3]**
   b. **echo [1-31-31-3]**
   c. **echo [0-9][0-9][0-9]**
   d. **echo [0-90-90-9]**
   e. **echo ???**

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

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

164. If the file **foo** contained the word **mom**, what would be the output on your screen of this two command sequence:
```
PATH=/bin/ps:/bin/echo:/bin/ls ; /bin/ls foo
```
a. **/bin/ls: foo: No such file or directory**
b. no output on screen
c. **mom**
d. **bash: /bin/ls: command not found**
e. **foo**

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

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

167. What is true about this output from **ls -il foo bar**
```
15 -r-x-----x 2 bin bin  3 Oct 30 09:23 foo
15 -r-x-----x 2 bin bin  3 Oct 30 09:23 bar
```
a. **foo** and **bar** are two of three names for this file
b. **foo** and **bar** each have three names (six names total)
c. **foo** and **bar** are names for the same file
d. this output is not possible
e. **foo** and **bar** are names for different files

168. If the file **foo** contained the word **mom**, what would be the output on your screen of this two command sequence:
```
PATH=/bin/ps:/bin/echo:/bin/ls ; /bin/ls foo
```
a. **mom**
b. no output
c. **/bin/ls: foo: No such file or directory**
d. **bash: /bin/ls: command not found**
e. **foo**

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

170. 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 [azAZ][azAZ][azAZ]**
b. **echo [a-zA-Za-zA-Za-zA-Z]**
c. **echo [a,zA,Z][a,zA,Z][a,zA,Z]**
d. **echo [a-zA-Z][a-zA-Z][a-zA-Z]**
e. **echo ???**

171. What is true about this output from **ls -ild foo bar**
```
96 -rwxr-xr-x 2 root root 3 Jan 24 01:03 foo
96 -rwxr-xr-x 3 root root 3 Jan 24 01:03 bar
```
a. this output is not possible
b. **foo** and **bar** are names for different files
c. **foo** and **bar** are two of five names for this file
d. **foo** and **bar** each have three names (six names total)
e. **foo** and **bar** are names for the same file

172. If **a=ant** and **b=bat** then what is the output on your screen of the following command sequence: **[$a = ant -a $b = ant] ; echo $?**
a. **0**
b. **test: $a: integer expression expected**
c. no output
d. **1**
e. the number 1 or 0 followed by another 1 or 0 on a new line

173. In an empty directory, what is the length of the longest file name created by the following two-command sequence:
```
var='1 12 123 1234 12345' ; touch '$var'
```
a. 2 characters
b. 19 characters
c. 4 characters
d. 5 characters
e. 3 characters

174. What is the output on your screen of the following command sequence:
```
i=0 ; test $i = 00 ; echo $?
```
a. no output
b. **0**
c. the number 0 or 1 followed by another 0 or 1 on a new line
d. **1**
e. **test: $i: integer expression expected**

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

176. If **a=ant** and **b=bat** then what is the output on your screen of the following command sequence: **[$a = bat -o $b = bat] ; echo $?**
   a. no output
   b. **0**
   c. **test: $a: integer expression expected**
   d. **1**
   e. the number 1 or 0 followed by another 1 or 0 on a new line

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

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

179. If **dog** is an executable script containing the line: **umask 0777** what is the output on your screen of the following sequence of commands:
   umask 0022 ; ./dog ; umask
   a. **0755**
   b. nothing; no output
   c. **0799**
   d. **0777**
   e. **0022**

180. What is the output on your screen of this two command sequence:
   PATH=/bin/ls:/bin/wc:/bin/sh ; wc nosuchfile
   a. **bash: wc: command not found**
   b. **bash: /bin/sh: No such file or directory**
   c. **bash: /bin/ls: command not found**
   d. **wc: nosuchfile: No such file or directory**
   e. **ls: /bin/wc: command not found**

181. Which command sequence correctly searches for **foo** and then prints the date if it is found inside the file **bar**?
   a. **if test foo bar ; then date ; fi**
   b. **if [grep foo bar] ; then date ; fi**
   c. **if [test foo bar] ; then date ; fi**
   d. **if grep <bar foo ; then date ; fi**
   e. **if test foo = bar ; then date ; fi**

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

183. What is the output on your screen of the following sequence of commands:
   x=0 ; [$x] ; echo $?
   a. the number 0 or 1 followed by another 0 or 1 on a new line
   b. **0**
   c. **test: $x: unary operator expected**
   d. **1**
   e. no output

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

185. If file **foo** contains nine lines, each of which is the number of the line in the file, what is the output on your screen of this command:
   cat foo foo | sort | tail -4 | head -1
   a. 1
   b. 4
   c. no output
   d. 8
   e. 6

186. If **/bin/pig** is a program that outputs **xx** and **/usr/bin/pig** is a program that outputs **foo** what is the output on your screen of this shell command sequence:
   PATH=/home:/bin:/dev:/usr/bin ; pig
   a. **bash: pig: command not found**
   b. **foo** followed by **xx**
   c. **xx**
   d. **xx** followed by **foo**
   e. **foo**

187. Which command sequence below outputs only lines 10-15 of the Unix password file?
   a. **head -15 /etc/passwd | tail -5 /etc/passwd**
   b. **tail -10 /etc/passwd | head -15 /etc/passwd**
   c. **tail -15 /etc/passwd | head -5**
   d. **head -15 /etc/passwd | tail -6**
   e. **head -10 /etc/passwd | tail -5 /etc/passwd**

188. If **bar** is an executable script containing the line **foo=dog** then what is the output on your screen 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**

189. 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?
a. Permissions **600** on directory **dir** and **700** on file **dir/bar**.
b. Permissions **200** on directory **dir** and **200** on file **dir/bar**.
c. Permissions **400** on directory **dir** and **400** on file **dir/bar**.
d. Permissions **100** on directory **dir** and **100** on file **dir/bar**.
e. Permissions **500** on directory **dir** and **600** on file **dir/bar**.

190. How many arguments and options are there to the command:
```
wc <infile -wc >wc
```
a. Three arguments, each of which is a pathname argument.
b. One argument: a single option argument with two option letters.
c. Two arguments: an input file and an option argument with two options.
d. Three arguments: two file names and one option argument with two options.
e. Four arguments, only one of which is an option argument with two options.

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

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

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

194. What is the link count of directory **d** after this set of successful commands?
```
mkdir d ; touch f ; cd d ; ln ../f x
```
a. 3
b. 5
c. 2
d. 1
e. 4

195. What is the output on your screen of this sequence of three shell commands:
```
umask 162 ; touch newfile ; ls -l newfile
```
a. **-rw---xr-x 1 me me 0 Oct 1 01:12 newfile**
b. **-rw----r-- 1 me me 0 Oct 1 01:12 newfile**
c. **----rw--w- 1 me me 0 Oct 1 01:12 newfile**
d. **-rw---x-w- 1 me me 0 Oct 1 01:12 newfile**
e. **---xrw--w- 1 me me 0 Oct 1 01:12 newfile**

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

197. A **Makefile** contains the following target: **foo: bar ant**
which means:
a. items bar and ant depend on item foo
b. items foo and bar depend on item ant
c. item foo depends on items bar and ant
d. the syntax "foo:" is not valid in a Makefile target
e. item ant depends on items foo and bar

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

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

200. What is the output on your screen of this two-command sequence:
```
cd /etc || echo "cd $(pwd)"
```
a. no output
b. `cd /etc`
c. `/etc`
d. `cd 0pwd)`
e. `cd $(pwd)`

201. Which command sequence correctly searches for `str` and then prints `OK` if it is found inside the file `foo`?
a. `if test str foo ; then echo OK ; fi`
b. `if [ grep str foo ] ; then echo OK ; fi`
c. `if [ test str foo ] ; then echo OK ; fi`
d. `if test str = foo ; then echo OK ; fi`
e. `if grep <foo str ; then echo OK ; fi`

202. What is the output on your screen of the following sequence of commands:
```
date='October Monday' ; test date = date
```
a. `test: too many arguments`
b. no output
c. `0`
d. `1`
e. `Mon Oct 27 17:01:38 EST 2003`

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

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

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

206. If `/bin/bat` is a program that outputs `foo` and `/usr/bin/bat` is a program that outputs `hi` what is the output on your screen of this shell command sequence:
```
PATH=/usr:/usr/bin:/bin ; bat
```
a. `foo` followed by `hi`
b. `foo`
c. `bash: bat: command not found`
d. `hi` followed by `foo`
e. `hi`

207. If `foo` is an executable script containing the line:
```
PATH=/bin ; export PATH
```
what is the output on your screen of the following sequence of commands:
```
PATH=/etc ; ./foo ; echo "$PATH"
```
a. `foo`
b. `$PATH`
c. `/bin`
d. `/etc`
e. `/etc:/bin`

208. In an empty directory, how many lines are in file `out` after this bash shell command line: `ls . .. nosuchfile 2>out`
a. `2`
b. `4`
c. `1`
d. no output (empty file)
e. `3`

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

210. Which of these commands makes a file owned by me, also readable by me?
a. `umask 300 ./myfile`
b. `chmod r+u myfile`
c. `chmod r=u ./myfile`
d. `umask 400 myfile`
e. `chmod u+r ./myfile`

211. What is the output on your screen of this command sequence:
```
true && echo Hello     There $?
```
a. `Hello There ?`
b. `Hello There 0`
c. `Hello     There ?`
d. `Hello     There 1`
e. no output

212. Which command line shows just the count of lines in the file?
   a. `wc file | awk '[print #1]'`
   b. `wc file | awk '[print $1]'`
   c. `wc file | awk '{print $1}'`
   d. `wc file | awk '{print 1}'`
   e. `wc file | awk '{print #1}'`

213. If `/bin/foo` is a program that outputs `mom` and `/usr/bin/foo` is a program that outputs `dad` what is the output on your screen of this shell command sequence:
```
PATH=/bin/foo:/usr/bin/foo:/usr ; foo
```
   a. `mom` followed by `dad`
   b. `dad`
   c. `dad` followed by `mom`
   d. `bash: foo: command not found`
   e. `mom`

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

215. If a 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 $2 $2`
   b. `./foo "bar" 'bar'`
   c. `./foo "$1" '$2'`
   d. `./foo bar bar`
   e. `./foo '$2' bar`

216. 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 ???`
   b. `echo [a-mn-zA-YZ][ab-zAB-YZ][za-yZA-Y]`
   c. `echo [azAZ][azAZ][azAZ]`
   d. `echo [0-89][01-9][0-45-9]`
   e. `echo [a,zA,Z][a,zA,Z][a,zA,Z]`

217. If `a=aaa` and `b=bbb` then what is the output on your screen of the following command sequence: `if $a = $b ; then echo $a ; fi`
   a. no output
   b. `test: aaa: integer expression expected`
   c. `aaa`
   d. `test: $a: integer expression expected`
   e. `bash: aaa: command not found`

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

219. If `a=1` and `b=1`, which 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 test a -eq b ; then echo OK ; fi`
   d. `if [ a = b ] ; then echo OK ; fi`
   e. `if [ $a==$b ] ; then echo OK ; fi`

220. In an empty directory, what is the output on your screen of these commands:
```
touch uu .u uv .v uw ; a="*u *v" ; echo "$a"
```
   a. `*u *v`
   b. `uu uv`
   c. `uu .u uv .v`
   d. `$a`
   e. `u* v*`

221. If `foo` is a script containing the line `TERM=linux ; export TERM` what is the output on your screen of the following sequence of commands:
```
TERM=vt100 ; ./foo ; echo "$TERM"
```
   a. `$TERM`
   b. `linux`
   c. `TERM`
   d. `vt100`
   e. `foo`

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

223. If `a=xxx` and `b=yyy` then what is the output on your screen of the following sequence of commands: `if $a = $b ; then echo $a ; fi`
   a. `xxx`
   b. `test: xxx: integer expression expected`
   c. no output
   d. `bash: xxx: command not found`
   e. `test: $a: integer expression expected`

224. Which command line shows just the type and permissions of file **foo**?
   a. **ls -l foo | awk ' ' '\n' | head -1**
   b. **cat foo | ls -l | awk ' ' '\n' | head -1**
   c. **ls -l foo | awk ' ' '\n'**
   d. **tr ' ' '\n' <ls -l foo | head -1**
   e. **ls -l foo | tr ' ' '\n' | head -1**

225. What appears on your screen after this sequence of commands:
```
echo 1 >x ; cp x y ; echo 2 >>y ; sort x >y ; cat y
```
   a. **1**
   b. **1** followed by **2**
   c. empty file - no output on the screen
   d. **2** followed by **1**
   e. **2**

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

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

228. In a directory containing one file named **dog**, what appears on your screen after this command line? **2>/dev/null ls nosuchfile**
   a. **ls: nosuchfile: No such file or directory**
   b. **dog**
   c. **bash: 2>/dev/null: command not found**
   d. **nosuchfile**
   e. no output on screen

229. What is the output on your screen of this sequence of three shell commands:
```
umask 475 ; mkdir dir ; ls -ld dir
```
   a. **dr--rwxr-x 2 it it 400 Jul 3 8:00 dir**
   b. **d-w-----w- 2 it it 400 Jul 3 8:00 dir**
   c. **d-wxrwx-w- 2 it it 400 Jul 3 8:00 dir**
   d. **d-wx----w- 2 it it 400 Jul 3 8:00 dir**
   e. **dr-xrwxr-x 2 it it 400 Jul 3 8:00 dir**

230. 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?
   a. Permissions **500** on directory **dir** and **500** on file **dir/bar**.
   b. Permissions **100** on directory **dir** and **300** on file **dir/bar**.
   c. Permissions **300** on directory **dir** and **400** on file **dir/bar**.
   d. Permissions **100** on directory **dir** and **500** on file **dir/bar**.
   e. Permissions **300** on directory **dir** and **200** on file **dir/bar**.

231. What is the output on your screen of the following sequence of commands:
```
a=pig ; b=bat ; touch $b ; test -n $b ; echo $?
```
   a. **test: $b: integer expression expected**
   b. **0**
   c. no output
   d. **1**
   e. the number 0 or 1 followed by another 0 or 1 on a new line

232. What is the correct syntax to redirect both standard output and standard error into the same output file?
   a. **command 2>out >out**
   b. **command 2>1 >out**
   c. **command >out 2>&1**
   d. **command 2>&1 >out**
   e. **command >out 2>1**

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

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

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

236. Which command line shows the file in **/bin** with the largest checksum?
   a. **sum /bin | sort -nr | head -1**
   b. **sum /bin/* | sort -nr | head -1**
   c. **ls /bin/* | sum | sort -nr | head -1**
   d. **cat /bin/* | sum | sort -nr | head -1**
   e. **cat /bin | sum | sort -nr | head -1**

237. If your **PATH** variable contains **/bin:/usr/bin**, what is the output on your screen of this command line: **echo '$PATH'**
   a. **$PATH**
   b. **echo: $PATH: No such file or directory**
   c. **/bin:/usr/bin**
   d. **'/bin:/usr/bin'**
   e. **'$PATH'**

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

239. What is the output on your screen of this two-command sequence:
   cd /etc/passwd && echo "in $(pwd)"
   a. **in $(pwd)**
   b. **bash: cd: /etc/passwd: Not a directory**
   c. **in 0pwd)**
   d. no output
   e. **in /etc**

240. Which command sequence below always outputs just the date only if the first argument is both not empty and a directory?
   a. **if [-s && -d "$1"]; then date ; fi**
   b. **if [-d "$1" -a -s "$1"]; then date ; fi**
   c. **if ["$1" -eq -f -a "$1" -eq -d]; then date ; fi**
   d. **if [-s -a -d "$1"]; then date ; fi**
   e. **if ["-s $1" && "-d $1"]; then date ; fi**

241. The correct option to enable warning messages from the **g++** compiler is:
   a. **+Warn**
   b. **-warn**
   c. **-wALL**
   d. **-Wall**
   e. **-wall**

242. 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 $2 $3**
   b. **./foo $1 '$2' $3**
   c. **./foo '$1' "$3" $2**
   d. **./foo $3 "$2" $1**
   e. **./foo 2 '$3' 1**

243. Which command stops people from using **write** to put lines of text on your screen?
   a. **write n**
   b. **mesg n**
   c. **chmod 000**
   d. **stop 0**
   e. **umask 000**

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

245. 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**

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

247. What is the output on your screen of the following sequence of commands:
   a=sky ; touch $a ; test -z $a ; echo $?
   a. **1**
   b. **test: $a: integer expression expected**
   c. **sky**
   d. **0**
   e. no output

248. What is the output on your screen of the following sequence of commands:
```
false && echo "linux     rocks $?"
```
a. no output
b. **linux rocks 1**
c. **linux rocks 0**
d. **linux rocks 1**
e. **linux rocks 0**

249. What can you do to get back (redo) the last command you typed to the **bash** (Linux) shell?
a. Use the "UpArrow" key.
b. Type [CONTROL]-[ALT]-[DEL]
c. Type [CONTROL]-[BACKSPACE]
d. Type [ALT]-[F2]
e. Use the "PageUp" key.

250. What is the link count of directory **d** after this set of successful commands?
```
mkdir d ; cd d ; touch a ; mkdir b c d
```
a. 3
b. 2
c. 5
d. 6
e. 4

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

252. Which command sequence correctly compares the numbers and prints **OK**?
a. **if [! 2 < 1] ; then echo OK ; fi**
b. **if (let 2 > 1) ; then echo OK ; fi**
c. **if [2 > 1] ; then echo OK ; fi**
d. **if [1 -lt 2] ; then echo OK ; fi**
e. **if (1 let 2) ; then echo OK ; fi**

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

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

255. What is the output on your screen of the following sequence of commands:
```
x=ok ; y=ok ; [ x = y ]
```
a. no output
b. **0**
c. **bash: x: command not found**
d. **1**
e. **test: x: integer expression expected**

256. What is the output on your screen of the following command sequence:
```
echo hi >wc ; wc wc >hi ; cat hi
```
a. **1 1 2 wc**
b. **0 0 0 wc**
c. **hi**
d. no output
e. **1 1 3 wc**

257. What is the output on your screen if a user signals an end-of-file from the keyboard during this command sequence? **read input ; echo $?**
a. no output on screen
b. **0**
c. **$?**
d. **1**
e. an error message

258. Which command line shows the current date?
a. **bash <date**
b. **date | bash**
c. **bash >date ; cat date**
d. **bash date**
e. **echo date | bash**

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

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

261. If **`foo`** is an executable script containing the line:
```
PATH=/etc ; export PATH
```
what is the output on your screen of the following sequence of commands:
```
PATH=/bin ; ./foo ; echo "$PATH"
```
a. **`$PATH`**
b. **`/etc`**
c. **`/bin`**
d. **`/bin:/etc`**
e. **`./foo`**

262. What is the output on your screen of this command sequence:
```
false && echo "hello     there $?"
```
a. **`hello there 0`**
b. **`hello     there 0`**
c. **`hello there 1`**
d. **`hello     there 1`**
e. no output

263. What is the link count of directory **`d`** after this set of successful commands?
```
mkdir d ; mkdir d/a ; touch d/b
```
a. 4
b. 1
c. 5
d. 3
e. 2

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

265. If **`bar`** is an executable script containing the line:
```
TERM=vt100 ; export TERM
```
what is the output on your screen of the following sequence of commands:
```
TERM=linux ; ./bar ; echo "$TERM"
```
a. **`$TERM`**
b. **`vt100`**
c. **`linux`**
d. **`./bar`**
e. **`TERM`**

266. What is the output on your screen of this sequence of three shell commands:
```
umask 457 ; mkdir newdir ; ls -ld newdir
```
a. **`d-wx-w---- 2 me me 512 Oct 1 1:12 newdir`**
b. **`d-wx-w-rwx 2 me me 512 Oct 1 1:12 newdir`**
c. **`dr-xr-xrwx 2 me me 512 Oct 1 1:12 newdir`**
d. **`dr--r-xrwx 2 me me 512 Oct 1 1:12 newdir`**
e. **`d-w--w---- 2 me me 512 Oct 1 1:12 newdir`**

267. If **`a=1`** and **`b=1`**, which command sequence correctly compares the two numbers as equal and prints **`OK`**?
a. **`if [ b = a ] ; then echo OK ; fi`**
b. **`if [ $a==$b ] ; then echo OK ; fi`**
c. **`if [ a -eq b ] ; then echo OK ; fi`**
d. **`if test a == b ; then echo OK ; fi`**
e. **`if test $b -eq $a ; then echo OK ; fi`**

268. If **`cow=cow`** and **`pig=pig`** then which of the following command lines outputs only the date (and nothing else)?
a. **`test cow = cow && date`**
b. **`[cow -ne pig] || date`**
c. **`[!cow = pig] && date`**
d. **`test cow -ne pig && date`**
e. **`[cow!=cow] || date`**

269. If **`/bin/foo`** is a program that outputs **`one`** and **`/usr/bin/foo`** is a program that outputs **`two`**, what is the output on your screen of this command sequence:
```
PATH=/bin/ls:/home:/usr/bin/cat:/etc ; foo
```
a. **`bash: foo: command not found`**
b. **`two`** followed by **`one`**
c. **`two`**
d. **`one`** followed by **`two`**
e. **`one`**

270. What is the output on your screen of the following sequence of commands:
```
i=0 ; test $i = 00 ; echo $?
```
a. the number 0 or 1 followed by another 0 or 1 on a new line
b. **`test: $i: integer expression expected`**
c. no output
d. **`1`**
e. **`0`**

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

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

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

274. If a 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 1 "$1"**
c. **./foo bar '$1'**
d. **./foo 'bar' "bar"**
e. **./foo bar 'bar'**

275. In an empty directory, how many lines are in file **bar** after this command line:
```
ls . nosuchfile 1>bar
```
a. empty file (no data)
b. **4**
c. **2**
d. **1**
e. **3**

276. What is the link count of file **f** after this set of successful commands?
```
rm f ; touch f ; cp f x
ln f a ; ln x y ; ln a z ; ln x b
```
a. 3
b. 4
c. 5
d. 2
e. 6

277. Which of the following shell command lines removes all the names in the current directory that are exactly three letters (alphabetic) long (and nothing else)?
a. **rm [a-zA-Z,a-zA-Z,a-zA-Z]**
b. **rm ???**
c. **rm [3][3][3]**
d. **rm [a-zA-Z][a-zA-Z][a-zA-Z]**
e. **rm [azAZ][azAZ][azAZ]**

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

Answer Key - NET 2003 − Ian Allen − Winter 2005 - NET 2003 Practice Test

Office use only: 1 52 148 30 7 174 109 46 261 32 147 223 13 175 31 97 103 159 118 262 156 273 26 251 236 171 253 112 2 265 199 119 113 140 93 77 226 84 278 82 11 193 205 190 169 256 128 259 85 195 125 131 89 6 218 96 214 239 268 222 151 137 245 73 121 144 104 74 51 198 161 215 233 138 90 111 35 115 36 272 18 237 8 63 209 242 141 217 66 126 181 276 221 133 188 33 5 232 15 200 258 212 204 72 235 42 196 231 208 203 101 67 134 179 247 277 40 83 210 86 177 116 76 124 17 182 110 3 95 225 224 216 16 160 197 274 254 180 250 142 14 12 136 54 219 102 213 22 184 163 149 255 39 249 139 23 173 60 238 145 266 167 252 81 206 165 207 80 191 264 211 58 107 146 44 59 20 106 43 194 243 170 162 99 75 34 240 37 9 21 269 168 53 130 183 64 4 62 105 201 244 155 19 275 45 25 49 94 164 270 178 227 28 150 69 263 57 220 55 98 47 100 65 228 117 246 87 91 186 10 152 123 108 24 122 230 88 192 202 234 114 71 248 68 271 154 153 157 120 127 48 241 257 79 172 143 135 229 27 189 50 176 129 61 38 185 56 41 29 158 187 78 166 70 92 132 267 260

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

185. d
186. c
187. d
188. e
189. e
190. b
191. a
192. c
193. d
194. c
195. b
196. e
197. c
198. b
199. e
200. a
201. e
202. b
203. b
204. e
205. e
206. e
207. d
208. c
209. d
210. e
211. b
212. c
213. d
214. b
215. e
216. b
217. e
218. d
219. a
220. a
221. d
222. c
223. d
224. e
225. a
226. e
227. c
228. e
229. d
230. c
231. b
232. c
233. e
234. d
235. b
236. b
237. a
238. e
239. b
240. b
241. d
242. e
243. b
244. c
245. a
246. d
247. a
248. a
249. a
250. c
251. b
252. d
253. c
254. d
255. a
256. e
257. d
258. e
259. a
260. e
261. c
262. e
263. d
264. c
265. c
266. a
267. e
268. a
269. a
270. d
271. b
272. d
273. b
274. c
275. d
276. a
277. d
278. d


```
Count of a:  55  20%
Count of b:  43  15%
Count of c:  54  19%
Count of d:  67  24%
Count of e:  59  21%

With 5 choices: 278
  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
 156 157 158 159 160 161
 162 163 164 165 166 167
 168 169 170 171 172 173
 174 175 176 177 178 179
 180 181 182 183 184 185
 186 187 188 189 190 191
 192 193 194 195 196 197
 198 199 200 201 202 203
 204 205 206 207 208 209
 210 211 212 213 214 215
 216 217 218 219 220 221
 222 223 224 225 226 227
 228 229 230 231 232 233
 234 235 236 237 238 239
 240 241 242 243 244 245
 246 247 248 249 250 251
 252 253 254 255 256 257
 258 259 260 261 262 263
 264 265 266 267 268 269
 270 271 272 273 274 275
 276 277 278

Macro .cmd split no indent: 5
Macro .cmd split with indent: 135
Macro .ans splits: 0
```