Evaluation: Part I - 55 M/C Questions **Name:** ________________________

Important Instructions

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

Multiple Choice - 55 Questions - 12 of 25%

(Office use only: 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)

1. 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 **500** on directory **dir** and **600** on file **dir/bar**.
   b. Permissions **100** on directory **dir** and **100** on file **dir/bar**.
   c. Permissions **200** on directory **dir** and **200** on file **dir/bar**.
   d. Permissions **400** on directory **dir** and **400** on file **dir/bar**.
   e. Permissions **600** on directory **dir** and **700** on file **dir/bar**.

2. 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 **300** on directory **dir** and **400** on file **dir/bar**.
   b. Permissions **100** on directory **dir** and **300** on file **dir/bar**.
   c. Permissions **100** on directory **dir** and **500** on file **dir/bar**.
   d. Permissions **300** on directory **dir** and **200** on file **dir/bar**.
   e. Permissions **500** on directory **dir** and **500** on file **dir/bar**.

3. 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. **var1=one var2=two var3=three**
   c. **one,two,three**
   d. **one:two:three**
   e. **$var1="one" $var2="two" $var3="three"**

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

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

6. 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 "$TERM"
   † **a.** **vt100**
   b. **linux**
   c. **foo**
   d. **TERM**
   e. **$TERM**

7. If **bat=1** and **cat=2** then which of the following command lines outputs only the word **foo** (and nothing else)?
   † **a.** **[bat = bat] && echo foo**
   b. **[bat -ne cat] && echo foo**
   c. **[!bat = cat] && echo foo**
   d. **[bat -eq 1] || echo foo**
   e. **[bat!=bat] || echo foo**

8. 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.** **0022**
   b. **0777**
   c. **0799**
   d. **0755**
   e. nothing; no output on screen

9. 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 -n | tail -1**
   b. **cat sort foo | uniq -c | sort -nr | head -1**
   c. **uniq -c foo | sort -nr | head -1**
   d. **sort | uniq -c | sort -n | tail -1 foo**
   e. **sort foo > uniq -c ; sort -nr uniq | head -1**

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

11. 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. **bash: cow: command not found**
 b. **test: cow: integer expression expected**
 c. **test: $a: integer expression expected**
 d. **cow**
 e. no output on screen

12. 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. **1**
 b. **0**
 c. the number 1 or 0 followed by another 1 or 0 on a new line
 d. **test: $a: integer expression expected**
 e. no output on screen

13. If a shell script named **foo** contains the line:
 if ['$3' = "$2"] ; then echo SAME ; fi
 then which of the following command lines will always produce **SAME** as output?
 † a. **./foo 1 '$3' 2**
 b. **./foo $1 '$2' $3**
 c. **./foo '$1' "$3" $2**
 d. **./foo $1 $2 $3**
 e. **./foo $3 "$2" $1**

14. 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. **/dir/????**
 b. **/dir/.123 /dir/.124 /dir/.???**
 c. **/dir/.123 /dir/.124**
 d. **echo: /dir/????: No such file or directory**
 e. no output on screen

15. 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 ; cp one ten
 † a. 8 blocks
 b. 6 blocks
 c. 4 blocks
 d. 2 blocks
 e. 10 blocks

16. 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 -eq :] ; then echo OK ; fi**
 c. **if [$x -eq ""] ; then echo OK ; fi**
 d. **if [''$x'' = ''''] ; then echo OK ; fi**
 e. **if ["$x" = *] ; then echo OK ; fi**

17. If **/bin/bat** is a program that outputs **mom** 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. **hi**
 b. **mom**
 c. **mom** followed by **hi**
 d. **hi** followed by **mom**
 e. **bash: bat: command not found**

18. 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. **one**
 c. **two**
 d. **two** followed by **one**
 e. **one** followed by **two**

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

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

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

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

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

24. In a directory containing one file named `foo`, what appears on your screen after this command line? `ls 2>/dev/null nosuchfile *`
   † **a.** `foo`
   b. no output on screen
   c. `nosuchfile foo`
   d. `ls: nosuchfile: No such file or directory`
   e. `ls: 2>/dev/null: No such file or directory`

25. In an empty directory, what is the length of the longest file name (including extension) after this sequence of commands?
   `ls >4444 ; cp 4444 55555 ; mv 55555 22 ; bzip2 22`
   † **a.** 6
   b. 5
   c. 4
   d. 3
   e. 7

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

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

28. 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. `2`
   d. `2` followed by `1`
   e. empty file - no output on the screen

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

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

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

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

33. What is the link count of directory `dir` after this set of successful commands?
   `mkdir dir ; cd dir ; touch a ; mkdir b c d e`
   † **a.** 6
   b. 5
   c. 4
   d. 3
   e. 7

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

35. 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 $?**
 † **a.** **1**
 b. no output on screen
 c. **$?**
 d. **0**
 e. an error message

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

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

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

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

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

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

42. What is the output on your screen of this two-command sequence if run in a directory containing 100 files with names that are all the numbers from **1** to **100** inclusive:
foo="*" ; echo $foo
 † **a.** the file names **1** through **100**
 b. all the file names that start with an asterisk ('*****')
 c. an asterisk ('*****') and the file names **1** through **100**
 d. *****
 e. **$foo**

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

44. What is true about this output from: **ls -il foo bar**
15 -r-x-----x 3 root root 3 Oct 30 09:23 foo
16 -r-x-----x 3 root root 3 Oct 30 09:23 foo
 † **a.** **foo** and **bar** are names for different files
 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** each have two names (four names total)
 e. this output is not possible

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

46. 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. **--x**
c. **r--**
d. **-wx**
e. **rw-**

47. What will appear on your screen if you execute this sequence of commands:
echo 1 >a ; ln a b ; echo 2 >b ; chmod 307 b ; cat a
† **a.** an error message
b. **1**
c. **2**
d. **1** followed by **2**
e. no output on screen

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

49. Which command line shows just the count of words in the file?
† **a.** **wc file | awk '{print $2}'**
b. **wc file | awk '{print #2}'**
c. **wc file | awk '{print 2}'**
d. **wc file | awk '[print $2]'**
e. **wc file | awk '[print #2]'**

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

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

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

53. 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 "$@"**
b. **cat "$*"**
c. **cat "$#"**
d. **cat "$1 $2 $3"**
e. **cat "$? $? $?"**

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

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

Answer Key - CST 8129 – Ian Allen – Fall 2005 - CST 8129 Unix Test #2 - 25%

Office use only: 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

1. a
2. a
3. a
4. a
5. a
6. a
7. a
8. a
9. a
10. a
11. a
12. a
13. a
14. a
15. a
16. a
17. a
18. a
19. a
20. a
21. a
22. a
23. a
24. a
25. a
26. a
27. a
28. a
29. a
30. a
31. a
32. a
33. a
34. a
35. a
36. a
37. a
38. a
39. a
40. a
41. a
42. a
43. a
44. a
45. a
46. a
47. a
48. a
49. a
50. a
51. a
52. a
53. a
54. a
55. a

Count of a: 55 100%

With 5 choices: 55

Macro .cmd split no indent: 1
Macro .cmd split with indent: 26