Evaluation: 40 Questions **Name:** _____

Important Instructions

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

One Mark Correct Answers are worth one mark each.

Zero Marks Unanswered questions are worth zero marks.

Minus 0.25 Marks Incorrect Answers are worth –0.25 (minus one quarter) mark each. Incorrect guesses will be partially subtracted from your overall score.

(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)

1. In the output of the command **`ls -a`**, a dot that begins a name signifies what?
† **a**. A name that is hidden.
b. The parent directory.
c. The current directory.
d. A current file.
e. A name with an unprintable character.

2. If you type the command **`cat`**
which of the following key sequences will send an EOF and take you immediately back to the command prompt?
† **a**. **`[CTRL-D]`**
b. **`[CTRL-C]`**
c. **`[CTRL-I]`**
d. **`[CTRL-U]`**
e. **`[CTRL-R]`**

3. Which command below removes *only* this five-character file name containing a special character (and no others): **`date?`**
† **a**. **`rm ./date\?`**
b. **`rm date/?`**
c. **`rm date\\?`**
d. **`rm date\*`**
e. **`rm ./date?`**

4. Which command below removes *only* this five-character file name containing a special character (and no others): **`*test`**
† **a**. **`rm "*test"`**
b. **`rm ./*test`**
c. **`rm \\*test`**
d. **`rm ''*test''`**
e. **`rm ./\\*test`**

5. How many arguments are there to this **`echo`** command:
`echo " one '2 three' 4 "five 6 ' 7 "8 ' >out`
† **a**. Three arguments.
b. Four arguments.
c. Five arguments.
d. Six arguments.
e. Two arguments.

6. If **`cow`** is a sub-directory that contains only the file **`dog`**, what happens after this command: **`mv cow/dog cow/./.cat`**
† **a**. the directory **`cow`** now contains only a file named **`cat`**
b. the directory **`cow`** is now empty
c. there is a second copy of the file **`dog`** in the file named **`cat`**
d. the command fails because the name **`cat`** does not exist
e. the command fails because the name **`cow/./.cat`** does not exist

7. If I have a directory owned by me named **`/a/b/c/7`**, which of the following actions would increase its *link count* by exactly one?
† **a**. create one subdirectory named **`/a/b/c/7/d2`**
b. create one subdirectory named **`/a/b/c/d/e`**
c. create one subdirectory named **`/a/b/c/7e`**
d. create one file named **`/a/b/c/7/d2`**
e. create one file named **`/a/b/c/7de`**

8. Which of the following is true, given this long directory listing:
`drwxr-x--x 512 ian user 712 May 30 12:35 dir`
† **a**. The number 512 is the count of links (names) this directory has.
b. The number 512 is the inode number of this directory.
c. The number 512 is the size of this directory.
d. The number 712 is the count of links (names) this directory has.
e. The number 712 is the inode number of this directory.

9. Given an existing file of yours named **`sort`**, what is the output of this sequence of three shell commands:
`echo hi >sort ; wc sort >sort ; cat sort`
† **a**. **`0 0 0 sort`**
b. **`1 1 3 sort`**
c. **`1 1 2 sort`**
d. **`2 2 4 sort`**
e. no output

10. Given an existing file of yours named **`wc`**, what is the output of this sequence of three shell commands: **`echo hi >wc ; sort wc >wc ; cat wc`**
† **a**. no output
b. **`0 0 0 wc`**
c. **`1 1 3 wc`**
d. **`1 1 2 wc`**
e. **`2 2 4 wc`**

11. How many arguments are passed to the command by the shell on this command line: **`<foo foo –foo >foo " foo >foo "foo`**

† **a**. **2**
b. **3**
c. **4**
d. **5**
e. **6**

12. Which of the following shell command lines displays only the names in the current directory that are exactly three digits (numeric) long?

† **a**. **`echo [0–89][01–9][0–45–9]`**
b. **`echo [a–zA–Z][a–zA–Z][a–zA–Z]`**
c. **`echo [1–3][1–3][1–3]`**
d. **`echo [?][?][?]`**
e. **`echo ???`**

13. If directory **`/a`** contains *only* these seven two-character file names: **`ad`**, **`bd`**, **`cd`**, **`dd`**, **`a.`**, **`a?`**, **`a*`**, then which shell command below will remove only the single name **`ad`** from the directory (and no others)?

† **a**. **`rm /a/a[a–z]`**
b. **`rm /a/a\?`**
c. **`rm /a/a?`**
d. **`rm /a/a*`**
e. **`rm /a/[a–z]d`**

14. If directory **`/a/b`** contains these four three-character file names: **`.aa`**, **`.ab`**, **`.a?`**, **`.a*`**, then what is the output of the following **`bash`** shell command line: **`echo /a/b/???`**

† **a**. **`/a/b/???`**
b. **`/a/b/.aa /a/b/.ab /a/b/.a? /a/b/.a*`**
c. **`/a/b/.aa /a/b/.ab`**
d. **`/a/b/.a?`**
e. no output

15. What is the **`bash`** shell output of this two command sequence:

```
PATH=/bin:/usr/bin ; cat nosuchfile
```

† **a**. **`cat: nosuchfile: No such file or directory`**
b. **`bash: cat: command not found`**
c. **`bash: /bin/cat: command not found`**
d. **`bash: /bin:/usr/bin: No such file or directory`**
e. **`cat: /bin:/usr/bin: No such file or directory`**

16. Which line below passes one *single* argument to the **`sort`** command when placed inside a shell script named **`foo`** invoked by the command line:

```
./foo a b c
```

† **a**. **`sort "$*"`**
b. **`sort "$@"`**
c. **`sort $*`**
d. **`sort $@`**
e. **`sort "$1" "$2" "$3"`**

17. Given the following **`bash`** shell command line: **`read a b c`**, which user keyboard input line below will assign the text **`y`** to the shell variable named **`b`**?

† **a**. **`x y z`**
b. **`a b c`**
c. **`x=a y=b z=c`**
d. **`a,b,c`**
e. **`x,y,z`**

18. What is the **`bash`** shell output of this two-command sequence if run in a directory containing 9999 files with names that are all the numbers from **`1`** to **`9999`** inclusive: **`abc='* *' ; echo '$abc'`**

† **a**. **`$abc`**
b. **`* *`**
c. **`'$abc'`**
d. the file names **`1`** through **`9999`**
e. the file names **`1`** through **`9999`**, repeated twice

19. What is the **`bash`** shell output of this two-command sequence if run in a directory containing 9999 files with names that are all the numbers from **`1`** to **`9999`** inclusive: **`var="*1" ; echo "$var"`**

† **a**. **`*1`**
b. **`$var`**
c. all the file names that end in the digit **`1`**
d. the file names **`1`** through **`9999`** followed by a digit **`1`**
e. all the file names that start with an asterisk (’**`*`**’)

20. What is the **`bash`** shell output of this two-command sequence if run in a directory containing 9999 files with names that are all the numbers from **`1`** to **`9999`** inclusive: **`foo="1*" ; echo $foo`**

† **a**. all the file names starting with the digit **`1`**
b. all the file names that end with an asterisk (’**`*`**’)
c. all the file names **`1`** through **`9999`**
d. **`1*`**
e. **`$foo`**

21. Which of these statements is true?
† **a**. If **/x** is an empty directory, **echo /x/*** produces no error message.
b. If **/y** is an empty directory, **cat /y/*** produces no error message.
c. Typing **./foo** and just **foo** always give identical results.
d. The **sort** command looks up a command line file name argument (i.e. **sort filename**) in your **$PATH**.
e. Quotes inside the text from expanded variables are special to the shell

22. What would you type to change the permissions on a file to **--x-wx-w-**?
† **a**. **chmod 132 file**
b. **chmod 121 file**
c. **chmod 322 file**
d. **chmod 122 file**
e. **chmod 654 file**

23. Which **bash** command line below turns *off* the **noclobber** option?
† **a**. **set +o noclobber**
b. **set -o noclobber**
c. **set +noclobber**
d. **set -noclobber**
e. **set noclobber=1**

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

25. If file **bar** contains the line **var=foo** then what is the **bash** output of this sequence of three commands:
```
var=abc ; read bar ; echo "output '$var' now"
```
† **a**. **output 'abc' now**
b. **output 'foo' now**
c. **output '$var' now**
d. **output $var now**
e. **output 'bar' now**

26. What is the **bash** output of this sequence of two commands:
```
semi=";" ; echo date $semi echo hi
```
† **a**. **date ; echo hi**
b. **Mon Oct 21 08:00:00 2002 ; hi**
c. **Mon Oct 21 08:00:00 2002 ; echo hi**
d. **Mon Oct 21 08:00:00 2002** followed by **hi** on a new line
e. **Mon Oct 21 08:00:00 2002** followed by **echo hi** on a new line

27. A shell script named **bar** is executed as follows:
```
./bar "one two" "three four five"
```
Inside the script is the line: **echo "$@"**
How many arguments are passed to the **echo** command inside the script?
† **a**. **2**
b. **1**
c. **3**
d. **4**
e. **5**

28. A shell script named **srt** is executed as follows:
```
./srt 1 " 23" "4 5"
```
Inside the script is the line: **head $***
How many arguments are passed to the **head** command inside the script?
† **a**. **4**
b. **5**
c. **3**
d. **1**
e. **2**

29. A shell script named **luk** is executed as follows:
```
./luk "a"'b' 'c d' e\ f
```
Inside the script is the line: **echo "$2"**
What is the output from this line?
† **a**. **c d**
b. **b**
c. **'b'**
d. **'c**
e. **e f**

30. What is the **bash** output of this command sequence:
```
false || echo foo bar $?
```
† **a**. **foo bar 1**
b. **foo bar 0**
c. **foo bar 1**
d. **foo bar 0**
e. no output

31. What is the **bash** shell output of this command sequence:
```
true && echo "one two $?"
```
† **a**. **one two 0**
b. **one two 1**
c. **one two 0**
d. **one two 1**
e. no output

32. What is the **bash** shell output of this two-command sequence:
```
cd /bin || echo "$(pwd) string"
```
† **a**. no output
b. **/bin string**
c. **0pwd) string**
d. **$(pwd) string**
e. **bash: cd /bin: No such file or directory**

33. What is the **bash** shell output of this two-command sequence:
```
cd /usr/bin && echo "the '$(pwd)' string"
```
† **a**. **the '/usr/bin' string**
b. **the '$(pwd)' string**
c. **the $(pwd) string**
d. **the '0pwd)' string**
e. no output

34. What is the **bash** shell output of this two-line command sequence:
```
foo () { echo "a '$1' and '$2' str" ; }
foo "1 2" '3' '4' '5 6'
```
† **a**. **a '1 2' and '3' str**
b. **a '$1' and '$2' str**
c. **a $1 and $2 str**
d. **a '1' and '2' str**
e. **a 1 2 and 3 4 str**

35. In an empty directory, what is the **bash** shell output of this three-command sequence: **touch .aa .ab .ac .ad ; x='???' ; echo '$x'**
† **a**. **$x**
b. **???**
c. **.aa .ab .ac .ad**
d. **??? ??? ??? ???**
e. no output

36. In an empty directory, what is the **bash** shell output of this three-command sequence:
```
touch .1 10 .2 21 2 33 ; x='1? 2*' ; echo "$x"
```
† **a**. **1? 2***
b. **$x**
c. **10 21 2**
d. **.1 10 .2 21 2**
e. **?1 *2**

37. In an empty directory, what is the **bash** shell output of this three-command sequence:
```
touch .1 10 11 .2 a1 22 a2 ; x='?1 *2' ; echo $x
```
† **a**. **11 a1 22 a2**
b. **.1 11 a1 .2 22 a2**
c. **$x**
d. **?1 *2**
e. **1? 2***

38. In an empty directory, how many files will be created using the following **bash** shell two-command sequence:
```
x="the  rat  'ate  bat'" ; touch $x
```
† **a**. 4 files
b. 1 file
c. 2 files
d. 3 files
e. 5 files

39. In an empty directory, how many files will be created using the following **bash** shell two-command sequence:
```
x='one  "day  cat"  dog' ; touch '$x'
```
† **a**. 1 file
b. 2 files
c. 3 files
d. 4 files
e. 5 files

40. In an empty directory, what is the length of the longest file name created by the following **bash** shell two-command sequence:
```
x='a bb ccc' ; touch "$x"
```
† **a**. 8 characters
b. 9 characters
c. 2 characters
d. 3 characters
e. 4 characters

Answer Key - CST 8129 – Ian Allen – Fall 2002 - CST 8129 Test #1-B - Unix - 10%

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

```
  1. a                          Count of letter a:  40  100%
  2. a
  3. a                          Questions with 5 choices: 40
  4. a                          1 2 3 4 5 6 7 8 9 10 11 12
    5. a                        13 14 15 16 17 18 19 20 21
    6. a                        22 23 24 25 26 27 28 29 30
    7. a                        31 32 33 34 35 36 37 38 39
    8. a                        40
  9. a
 10. a                          Macro .cmd splits: 19
 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
```