

Evaluation: 57 Questions

Name: \_\_\_\_\_

**Important Instructions**

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

(Office use only: 13 30 26 9 25 54 51 42 40 43 47 38 4 17 50 14 7 19 55 52 46 36 8 45 21 2 15 28 57 35 49 48 16 53 11 39 1 31 56 37 12 24 41 44 20 29 22 33 34 3 18 32 5 27 6 10 23)

1. If **a=1** and **b=2** then which of the following **bash** command lines outputs only the word **hi** (and nothing else)?
  - a. `[ !a = b ] && echo hi`
  - b. `[ a!=a ] || echo hi`
  - c. `[ a -ne b ] && echo hi`
  - d. `[ a = a ] && echo hi`
  - e. `[ a -ne b ] || echo hi`
2. If **x=8** and **y=9** then which of the following **bash** command lines outputs only the word **foobar** (and nothing else)?
  - a. `[ x!=x ] || echo foobar`
  - b. `[ !x = y ] && echo foobar`
  - c. `[ x -ne y ] || echo foobar`
  - d. `[ x = x ] && echo foobar`
  - e. `[ x -ne y ] && echo foobar`
3. If **foo** is a script containing the line **TERM=new ; export TERM**, what is the output of the following sequence of **bash** commands that use **foo**:  
`TERM=bar ; ./foo ; echo $TERM`
  - a. **foo**
  - b. **TERM**
  - c. **\$TERM**
  - d. **new**
  - e. **bar**
4. 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-zA-Z][za-yZA-Y]`
  - c. `echo [azAZ][azAZ][azAZ]`
  - d. `echo [a,zA,Z][a,zA,Z][a,zA,Z]`
  - e. `echo [0-89][01-9][0-45-9]`

5. Which **bash** command sequence below always outputs just the word **OK** only if the first argument is either a file or a directory?
  - a. `if [ "$1" -eq -f -o "$1" -eq -d ]; then echo OK;fi`
  - b. `if [ -f || -d "$1" ]; then echo OK;fi`
  - c. `if [ -f -o -d "$1" ]; then echo OK;fi`
  - d. `if [ -f "$1" -o -d "$1" ]; then echo OK;fi`
  - e. `if [ "-f $1" || "-d $1" ]; then echo OK;fi`
6. In an empty directory, how many files will be created using the following **bash** shell two-command sequence:  
`x='12 3 4 5' ; touch $x`
  - a. 5 files
  - b. 4 files
  - c. 2 files
  - d. 3 files
  - e. 1 file
7. In an empty directory, what is the **bash** shell output of this three-command sequence:  
`touch aa .a ab .b ac .c ; x='*a *b' ; echo $x`
  - a. **\*a \*b**
  - b. **\$x**
  - c. **aa ab**
  - d. **aa ab ac b\***
  - e. **a\* b\***
8. If file **foo** contains the line **x=123** then what is the **bash** output of this sequence of three commands:  
`x=abc ; source foo ; echo "I see '$x' here."`
  - a. **I see \$x here.**
  - b. **I see '123' here.**
  - c. **I see 'abc' here.**
  - d. **I see '\$x' here.**
  - e. **"I see abc here."**
9. 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: `x="*" ; echo $x`
  - a. **\***
  - b. the file names **1** through **9999**
  - c. **\$x**
  - d. an asterisk (**\***) and the file names **1** through **9999**
  - e. all the file names that start with an asterisk (**\***)

10. What is the **bash** output of this sequence of two commands:  
`x=';' ; echo one $x date`
- `one ; date`
  - `one $x date`
  - `one` followed by `Mon Sep 30 08:00:00 EDT 2002` on a new line
  - `one ; Mon Sep 30 08:00:00 EDT 2002`
  - `one ';' date`
11. A shell script named **foo** is executed as follows:  
`./foo 1 "2 3 4" 5`  
 Inside the script is the line: `echo "$2"`  
 What is the output from this line?
- 2
  - "2
  - \$2
  - a bash error message: unbound (undefined) variable
  - 2 3 4
12. What is the **bash** shell output of this two-command sequence if run in a directory containing 9999 files with names that are all the numbers from 1 to 9999 inclusive: `x="*" ; echo '$x'`
- `$x`
  - \*
  - `'$x'`
  - the file names 1 through 9999
  - the file names 1 through 9999, surrounded by quotes
13. What is the output of the following sequence of **bash** commands:  
`false && echo "foo bar $?"`
- `foo bar 1`
  - `foo bar 1`
  - `foo bar 0`
  - no output
  - `foo bar 0`
14. If `a=cow` and `b=dog` then what is the output of the following sequence of **bash** commands: `[ $a = cow -a $b = cow ] ; echo $?`
- 0
  - 1
  - the number 1 or 0 followed by another 1 or 0 on a new line
  - `test: $a: integer expression expected`
  - no output

15. Select the correct **bash** shell order of command line processing:
- aliases, variables, redirection, globs
  - aliases, globs, variables, redirection
  - aliases, redirection, variables, globs
  - aliases, variables, globs, redirection
  - redirection, aliases, globs, variables
16. What is the output of the following sequence of **bash** commands:  
`a=1 ; b=2 ; test $a -ge $b ; echo $?`
- no output
  - the number 1 or 0 followed by another 1 or 0 on a new line
  - 1
  - `test: $a: integer expression expected`
  - 0
17. A shell script named **foo** is executed as follows:  
`./foo 1 "2 2 2" '3 '`  
 Inside the script is the line: `argv.sh $@`  
 What is the count of arguments that the `argv.sh` command will display?
- 6
  - 2
  - 4
  - 3
  - 5
18. If `a=cow` and `b=dog` then what is the output of the following sequence of **bash** commands: `if $a = $b ; then echo $a ; fi`
- no output
  - `test: $a: integer expression expected`
  - `test: cow: integer expression expected`
  - `cow`
  - `bash: cow: command not found`
19. In an empty directory, how many files will be created using the following **bash** shell two-command sequence:  
`x='1 2 3 45' ; touch "$x"`
- 5 files
  - 2 files
  - 4 files
  - 1 file
  - 3 files

20. In an empty directory, what is the **bash** shell output of this three-command sequence:

```
touch aa .a ab .b ac .c ; x='a* b*' ; echo "$x"
```

- \$x
- a\* b\*
- aa ab ac b\*
- aa ab
- \*a \*b

21. A shell script named **foo** is executed as follows:

```
./foo 1 "2 3 4" 5
```

Inside the script is the line: **argv.sh "\$@"**

What is the count of arguments that the **argv.sh** command will display?

- 4
- 5
- 3
- 1
- 2

22. 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 a b c
```

- sort "\$1 \$2 \$3"
- sort "\$#"
- sort "\$? \$? \$?"
- sort "\$@"
- sort "\$\*"

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

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

- hi
- 1 1 3 wc
- no output
- 1 1 2 wc
- 0 0 0 wc

24. A shell script named **foo** is executed as follows:

```
./foo a b "c d e"
```

Inside the script is the line: **argv.sh "\$\*"**

What is the count of arguments that the **argv.sh** command will display?

- 5
- 1
- 2
- 3
- 4

25. Which **bash** command sequence correctly compares the two numbers and prints **OK**?

- if [ 4 > 3 ] ; then echo OK ; fi
- if [ 4 -gt 3 ] ; then echo OK ; fi
- if ( let 4 > 3 ) ; then echo OK ; fi
- if ( ! 4 < 3 ) ; then echo OK ; fi
- if [ ! 4 <= 3 ] ; then echo OK ; fi

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

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

- in \$(pwd)
- in /etc
- bash: cd: /etc: No such file or directory
- no output
- in 0pwd)

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

```
a=cow ; b=dog ; test -z $a ; echo $?
```

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

28. Which line below puts the count of the number of lines in the password file into the variable **foo**?

- foo=[ wc /etc/passwd | echo \$1 ]
- foo=[ cat -l /etc/passwd ]
- foo=[ grep -c /etc/passwd ]
- foo=\$( cat -c /etc/passwd )
- foo=\$( wc -l </etc/passwd )

29. If these two lines are put in an executable script named **foo**:

```
#!/bin/cp bar
```

```
echo hi
```

What is the result of the command line: **./foo**

- The file **./foo** is copied to the file **bar**
- The file **bar** is copied to the file **./foo**
- The word "hi" appears on the screen
- The file **bar** appears on the screen followed by the word "hi"
- The **cp** command displays an error message about a missing argument

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

```
x=0 ; y=1 ; test ! -z $x ; echo $?
```

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

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

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

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

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

```
true && echo Hello      There $?
```

- Hello There 1
- no output
- Hello There ?
- Hello There ?
- Hello There 0

33. If **a=cow** and **b=dog** then what is the output of the following sequence of **bash** commands: [ **\$a = dog -o \$b = cow** ] ; **echo \$?**

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

34. In an empty directory, what is the **bash** shell output of this three-command sequence:

```
touch aa .a ab .b .c ; x='.a* .b*' ; echo '$x'
```

- '.a\* .b\*'
- .a .b
- .a\* .b\*
- aa .a ab .b
- \$x

35. Which of these commands makes a file owned by me, also executable by me?

- umask 777 myfile**
- chmod u+x ./myfile**
- umask 111 myfile**
- chmod x+u myfile**
- chmod x=u ./myfile**

36. 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: **x="\*" ; echo "\$x"**

- "\$x"
- the file names 1 through 9999, surrounded by quotes
- the file names 1 through 9999
- \$x
- \*

37. If **bar** is an executable script containing the line **foo=dog** then what is the **bash** output of this sequence of three commands:

```
foo=cat ; ./bar ; echo "the '$foo' ate"
```

- the 'cat' ate
- the 'dog' ate
- the \$foo ate
- the '\$foo' ate
- the 'foo' ate

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

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

- 1 1 2 wc
- 1 1 3 wc
- wc
- no output
- 0 0 0 wc

39. In an empty directory, what is the length of the longest file name created by the following **bash** shell two-command sequence:

```
x='1 12 123 1234' ; touch '$x'
```

- 3 characters
- 2 characters
- 4 characters
- 1 character
- 13 characters

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

- a:b:c**
- a,b,c**
- a;b;c**
- a b c**
- a=a b=b c=c**

41. What is the output of the following sequence of **bash** commands:
- ```
wc='one two' ; test wc = wc
```
- test: too many arguments**
  - 1**
  - 1 2 8 wc**
  - no output
  - 0**
42. If variable **foo** might contain nothing (a null value - defined but empty), which **bash** command sequence correctly tests for this and prints **OK**?
- if [ "\$foo" = \* ] ; then echo OK ; fi**
  - if [ \$foo -eq : ] ; then echo OK ; fi**
  - if [ "\$foo" = "" ] ; then echo OK ; fi**
  - if [ '\$foo' = '' ] ; then echo OK ; fi**
  - if [ \$foo -eq "" ] ; then echo OK ; fi**
43. Which **bash** command line below allows programs in the current directory to execute without preceding the names with **./**?
- PATH=/bin:\$HOME:.**
  - \$PATH=.:\$HOME:/bin**
  - \$PATH=/bin:./\$HOME**
  - PATH = /bin:\$HOME:.**
  - PATH = ./\$HOME:/bin**
44. What is the **bash** output of this command sequence:
- ```
false && echo "Hello There"
```
- Hello There**
  - "Hello There"**
  - HelloThere**
  - Hello There**
  - no output
45. If a **bash** shell script named **foo** contains the line:
- ```
if [ '$1' = "$2" ] ; then echo SAME ; fi
```
- then which of the following command lines will produce **SAME** as output?
- ./foo 1 "\$1"**
  - ./foo 'bar' "bar"**
  - ./foo \$1 \$1**
  - ./foo bar 'bar'**
  - ./foo bar '\$1'**

46. If **foo** is a script containing the line **TERM=vt100 ; export TERM**, what is the output of the following sequence of **bash** commands:
- ```
TERM=linux ; ./foo ; echo $TERM
```
- \$TERM**
  - linux**
  - vt100**
  - foo**
  - TERM**
47. If **x=5** and **y=5**, which **bash** command sequence correctly compares the two numbers as equal and prints **OK**?
- if ( x == y ) ; then echo OK ; fi**
  - if test \$x -eq \$y ; then echo OK ; fi**
  - if [ \$x==\$y ] ; then echo OK ; fi**
  - if test x -eq y ; then echo OK ; fi**
  - if [ x = y ] ; then echo OK ; fi**
48. A shell script named **bar** is executed as follows:
- ```
./bar a "b c" 'a '
```
- Inside the script is the line: **argv.sh \$@**
- What is the count of arguments that the **argv.sh** command will display?
- 2**
  - 3**
  - 4**
  - 6**
  - 5**
49. If variable **bar** might contain nothing (a null value - defined but empty), which **bash** command sequence correctly tests for this and prints **YO**?
- if [ '\$bar' = '' ] ; then echo YO ; fi**
  - if [ "\$bar" = "" ] ; then echo YO ; fi**
  - if [ \$bar -eq : ] ; then echo YO ; fi**
  - if [ "\$bar" = \* ] ; then echo YO ; fi**
  - if [ \$bar -eq "" ] ; then echo YO ; fi**
50. In an empty directory, how many files will be created using the following **bash** shell two-command sequence:
- ```
x="one 'two two'two three four" ; touch $x
```
- 3 files**
  - 1 file**
  - 4 files**
  - 2 files**
  - 5 files**

51. If **a=cow** and **b=dog** then what is the output of the following sequence of **bash** commands: `[ $a = dog -o $b = dog ] && echo $?`
- the number 1 or 0 followed by another 1 or 0 on a new line
  - 1
  - no output
  - 0
  - `test: $a: integer expression expected`
52. Which **bash** command sequence correctly searches for the string **foobar** and then prints **YES** if it is found inside the group file?
- `if test foobar /etc/group ; then echo YES ; fi`
  - `if test foobar = /etc/group ; then echo YES ; fi`
  - `if grep foobar /etc/group ; then echo YES ; fi`
  - `if [ test foobar /etc/group ] ; then echo YES ; fi`
  - `if [ grep foobar /etc/group ] ; then echo YES ; fi`
53. How many arguments are passed to the command by the shell on this command line: `<bar bar -b "-a" '-r' >bar bar bar`
- 2
  - 5
  - 3
  - 4
  - 6
54. Which line below is most likely to be the beginning of an error message?
- `echo 2>&1 "... "`
  - `echo 1<&2 "... "`
  - `echo 2<$1 "... "`
  - `echo 2>$1 "... "`
  - `echo 1>&2 "... "`
55. A shell script named **foo** is executed as follows:
- ```
./foo 1 "2 3 4" '5 '
```
- Inside the script is the line: `argv.sh "$@"`
- What is the count of arguments that the **argv.sh** command will display?
- 5
  - 6
  - 3
  - 2
  - 4

56. Which of these first lines will cause this executable file to be interpreted using the Bash shell?
- `#!/bin/bash`
  - `!/bin/bash`
  - `/bin/bash -u`
  - `#!/bin/bash`
  - `#!/bin/bash -u`
57. Which **bash** command sequence correctly searches for the **string** and then prints **OK** if it is found inside the password file?
- `if [ test string /etc/passwd ] ; then echo OK ; fi`
  - `if test string /etc/passwd ; then echo OK ; fi`
  - `if test string = /etc/passwd ; then echo OK ; fi`
  - `if grep string /etc/passwd ; then echo OK ; fi`
  - `if [ grep string /etc/passwd ] ; then echo OK ; fi`

**Answer Key - DAT 2330 – Ian Allen – Winter 2003 - DAT 2330 Test  
#3 - Practice Unix Final - 0%**

Office use only: 13 30 26 9 25 54 51 42 40 43 47 38 4 17 50 14 7 19 55 52 46 36 8 45 21 2 15 28 57 35 49 48 16 53 11 39 1 31 56 37 12 24 41 44  
20 29 22 33 34 3 18 32 5 27 6 10 23

- |       |                         |
|-------|-------------------------|
| 1. d  | 41. d                   |
| 2. d  | 42. c                   |
| 3. e  | 43. a                   |
| 4. b  | 44. e                   |
| 5. d  | 45. e                   |
| 6. b  | 46. b                   |
| 7. c  | 47. b                   |
| 8. b  | 48. c                   |
| 9. b  | 49. b                   |
| 10. a | 50. e                   |
| 11. e | 51. d                   |
| 12. a | 52. c                   |
| 13. d | 53. b                   |
| 14. b | 54. e                   |
| 15. c | 55. c                   |
| 16. c | 56. d                   |
| 17. e | 57. d                   |
| 18. e |                         |
| 19. d | Count of a: 5 9%        |
| 20. b | Count of b: 18 32%      |
| 21. c | Count of c: 9 16%       |
| 22. d | Count of d: 11 19%      |
| 23. b | Count of e: 14 25%      |
| 24. b |                         |
| 25. b | With 5 choices: 57      |
| 26. b | 1 2 3 4 5 6 7 8 9 10 11 |
| 27. a | 12 13 14 15 16 17 18 19 |
| 28. e | 20 21 22 23 24 25 26 27 |
| 29. b | 28 29 30 31 32 33 34 35 |
| 30. e | 36 37 38 39 40 41 42 43 |
| 31. c | 44 45 46 47 48 49 50 51 |
| 32. e | 52 53 54 55 56 57       |
| 33. b |                         |
| 34. e | Macro .cmd splits: 31   |
| 35. b | Macro .ans splits: 0    |
| 36. e |                         |
| 37. a |                         |
| 38. e |                         |
| 39. b |                         |
| 40. d |                         |