5

Part 1 (10 marks) **Each correct answer: 1 mark.**

Circle the letter corresponding to the one correct answer to each of the following questions.

1. What will this command do? **cat < xfile | grep xfile**

   a. pipes the standard input of cat to the standard output of grep
   [✓ b.] pipes the standard output of cat to the standard input of grep
   c. pipes the standard input of cat to the standard input of grep
   d. pipes the standard output of cat to the standard output of grep

2. Which line will **not** be matched by the following extended regular expression: **^[^a-c]*(cat|dog)?a+b$**

   a. tcatdogab
   [✓ b.] ab
   c. ipdogaaab
   d. qdogcataab

3. With a umask of 023, a directory called "testing" was created. Which command will ensure that **all** users can copy files to this directory?

   a. chmod u+wx,g+x testing
   [✓ b.] chmod go+w,o+x testing
   c. chmod o+w,go+x testing
   d. chmod g+wx,u+x testing

4. With a umask of 026, what will be the default permissions for newly-created ordinary files?

   [✓ a.] r w x r - x - - x
   b. - - - - w - r w -
   c. r w - r - - - - -
   d. r w x r - - - - -

   4 2 1

5. Which filename **will** be matched by the following extended glob: [!a-c]*(cat|dog)?ab

   a. tcatdogab
   [✓ b.] ab
   c. ipdogaaab
   d. qdogcataab

6. The variable containing the exit status of the previous command is:

[✓ a.] $?
b. $0
c. $*
d. $#

7. Failure of a command is usually indicated by:

a. an exit status of 0
b. an error message redirected to standard output
[✓ c.] a non-zero exit status
d. getting results different than expected

8. If script "zoo" is called as follows: **zoo lion tiger bear**
and "zoo" executes the following: **set 24 375 ; echo $1 $2 $3**
what will be displayed?

a. lion tiger bear
✗ [✓ b.] 24 375 bear
c. lion 24 375
→ d. 24 375

9. One method to read "file1" one line at a time in a loop is:

a. read line < file1 | while
→ b. cat file1 | while read line
✗ [✓ c.] for line in file1
d. while read line < file1

10. You need a file called "test2" which duplicates the contents of "test1", without deleting "test1". Which command **cannot** be used to meet this requirement?

a. cat
b. cp
c. grep
[✓ d.] mv

4

Part 2 (10 marks) Each correct answer: 1 mark.

Match the ***Symbol/Command*** number beside the best ***Explanation***.

Matched Number **Explanation**

(1) **(EXAMPLE)** Input redirection.

xargs
(~~17~~) 20 Command to convert piped input to parameters.

$0
(~~6~~) 11 Name by which shell script was called.

Ctrl-d
(~~2~~) 16 End-of-file for STDIN.

kill Ctrl-C
(8) Terminate a long-running command.

~~2>~~
(~~12~~) 17 Redirect command output to STDERR. 2>

$*
(7) All script arguments as a single string.

exec
(~~13~~) 5 Command to assign files as file descriptors.

~~cd~~ pwd
(10) Current directory.

||
(4) Execute a command if the previous one failed.

which
(~~5~~) 13 Command to display path of an executable.

Symbol / Command

~~1.~~	<
~~2.~~	file
3.	whereis
~~4.~~	||
~~5.~~	exec
~~6.~~	$$
~~7.~~	$*
~~8.~~	<Ctrl>-c
9.	<Ctrl>-z
~~10.~~	.
11.	$0
~~12.~~	2>
~~13.~~	which
14.	&&
15.	find
16.	<Ctrl>-d
~~17.~~	>&2
18.	$2
19.	..
20.	xargs

3.5

Part 3 (10 marks) Each correct answer: 1 mark.

Write the shortest single command, including any required options and pathnames, to answer the following questions. Do not assume a specific current directory.

1. Write a command to display character positions 10 to 20 in every line of a file called "personnel" in the current directory:

~~egrep -p[10-20] personnel~~

0.

cut -c10-20 personnel

2. Write a command to display "Enter your name" with no new-line on the terminal, regardless of any redirections specified when calling the containing script:

echo "Enter your name"

0

—

-n

echo -n "Enter your nam" > /dev/tty

3. Write a command to execute file "a.out" in the current directory, sending output to a file called "total.tax" in the "/tmp" directory:

1

a.out > /tmp/total.tax

4. Write a command to delete a directory called "dir1" from the current directory, including any contents, adding any error messages to the end of file "messages" in your home directory:

0.5

rm -r dir1 2>> ~/messages

5. Write a command to list all filenames in the current directory that consist of an 'a' followed by one or more digits, assuming extended globbing is enabled:

0

~~ls -a | egrep 'a[0-9]+'~~

ls -d a+([0-9])

6. Write a command to allow users in your group to add or delete files in the parent of your current directory, without changing any other permissions:

0.5 chmod g+~~x~~wx ..

7. Write a command to place the PID (process ID number) of the current process into the shell variable "$1":

0.5 set ~~$1 pd~~

set $$

8. Write a command to subtract 2 from the first argument passed to the script, and place the result into a variable called "number":

0 number=$(($1-~~$~~2))

((number = $1-2)) or

number = $(($1-2))

9. Write a command which will display **only** the 7th line in a file called "stock" in the current directory:

5 sed -n "7 p" stock

sed -n '7p' stock or

awk "NR==7" stock

10. Write a command to display **only** lines containing at least 1000 characters, in a file called "xfiles" in the current directory:

5 awk '-nc >=1000' xfiles

egrep ".{1000}" xfiles

2

Part 4 (10 marks) Each correct answer: 2 marks.

Write the shortest single UNIX command line required to perform each of the following tasks, without using the semi-colon (;) command separator. Do not assume a specific current directory. Efficiency counts, the fewer and simpler the commands and arguments, the better:

1. Display the contents of **all** files in the current directory that have names ending with ".txt", converting all alphabetic characters to uppercase.

0.5

ls -al | egrep '.txt$' | tr '[a-z]' '[A-Z]'

cat *.txt | tr '[a-z]' '[A-Z]'

2. Display the permissions for an ordinary file called "file1" in the current directory, in the following (example) format:
r w x r - x r - - (note the blanks between the characters).

0.5

ls -l file1 | ~~awk~~ '$1'

ls -l file1 | cut -c2-10 | sed 's/./&_/g'

anything follow w/ _ & append the match with _

3. Assign the number of fields in the file called "marks.dat" to the variable "fields". The number of fields is the same within each line of "marks.dat" and the fields are delimited by a space. The "marks.dat" file is contained in your current directory.

0.5

fields=$(wc -d' ' mark.dat)

fields=$(head -1 marks.dat | wc -w) or fields=$(awk 'NR==1{print NF}' marks.dat)

4. Here is a sample line from the output of an "ls -al" command:

```
-rw-r--r--   1 jbloe   users        1878 Mar 16 20:21 index.html
```

Display this information for the largest file in your current directory.

k - for field

ls -alS | head -1

.5

ls -al | ~~find -ls~~

or

ls -al | sort -nk 5 | tail -1

5. If the first script argument is a single-digit number, display that field number within the tab-delimited file "file2". For example, if the first argument is "7", then the 7th field in each record of "file2" should be displayed. For full marks, do not use "grep" and use a named class.

if test -n $1
then awk '$7' file2
fi

[[$1 = [[:digit:]]]] && cut -f$1 file2

8

Part 5 (10 marks)

My attempt at writing a script is shown below. It doesn't seem to work. Place an "X" beside each incorrect or missing line, and show the corrections required. There are several errors, please identify and correct any 5 of them.

```
if [ $# = 0 ]

   then set sysadmin 25

elif [ $# = 1 ]

   then echo $1 > file1

        if cat file1 | egrep '^[0-9][0-9]*$' file1 > /dev/null ✗
      ✓ if cat file1 | egrep '^[0-9][0-9]*$' > /dev/null
           then set $((whoami)) $1 ✗
      ✓    then set $(whoami) $1
           else set $1 25

        fi

elif [ $# > 2 ] ⟶ [$ -gt 2] or [[$# >2]]

✓  then "Usage: greatscript [ userid ] [ number ]" >&2 ✗
   then echo "Ucage: greatscript [$userid] [$number]" >
else if echo $1 | egrep '^[0-9][0-9]*$' > /dev/null
     then
        set $2 $1

fi
                          $2           ~~$~~$1
echo "The answer is ~~$1~~" | mail ~~$2~~   → you want to send
                                              the mail to username
✗
✓ fi

✗
exit 0
```

9

Part 6 (10 marks)

Here is a listing of directory **dir1** and the script **listfile:**

```
==> ls -al dir1
total 216
drwx------ 3 lczegel users   4096 Oct 29 22:00 .
drwx--x--x 5 lczegel users   4096 Oct 29 22:02 ..
-rwx------ 1 lczegel users 135838 Oct 10 11:29 .phones
-rwx------ 1 lczegel users   4178 Oct 10 11:29 phonebook.dat
drwx------ 4 lczegel users   4096 Oct 10 11:29 testdir1
-rwx------ 1 lczegel users  26615 Oct 10 11:29 vimtutor
==> cat listfile
if [ $# != 1 ]
   then echo "Message 1" >&2
        exit 3
fi
if [ ! -d $1 ]
   then echo "Message 2" >&2
        exit 4
fi
for filename in $(ls -a $1)
do
   if [ ! -d $1/$filename ]
   then
      echo $filename : $(cat $1/$filename | wc -c)
   fi
done
exit 8
==>
```

Show the result displayed upon entering each of the following:

==> listfile dir1 2>/dev/null; echo $?

.phones
phonebook.dat
vimtutor
8

==> listfile dir1/phonebook.dat >/dev/null; echo $?

Message 2
4.

==> listfile dir1 phonebook.dat 2>/dev/null; echo $?

3