Oregon Institute of Technology

CST 240 Linux Programming

Lab #4: Files/Directories, Bash Encryption

CST 240
LAB 4: Files/Directories, Bash Encryption

1. (20 pts) Files/Directories

Consider the script below:

```
#!/bin/bash
removelog="/home/student/remove.log"
if [ $# -eq 0 ] ; then
    echo "Usage: $0 [-s] list of files or directories" >&2
    exit 1
fi

if [ "$1" = "-s" ] ; then
    shift
else
    echo "$(date): ${USER}: $@" >> $removelog
fi

/bin/rm "$@"
exit 0

##############################################
```

Figure out what the script above does. Execute it all the ways you can think of.
Hint: do `$ touch a.txt b.txt c.txt d.txt`
Then test out the script on the files you just created. Be sure to test boundary conditions and unexpected data.

Describe what this program does.
Describe the following:

- What is the input including the boundaries (what isn't acceptable)
- What is the output
- How is this transformation achieved, giving specifics about what each if block does

This should be about 4 paragraphs with the 1st, 3rd and 4th paragraphs being about 3-5 sentences each, and the 2nd paragraph being about 1 sentence.

The expected input is the file(s) to be removed, with or without a -s. The output is the removal of the file, and, if -s is NOT used, then a file is either created or appended with information about the removed file. If the script is called without a file sent (no input), then the output to the terminal is usage information. If a file that doesn't exist is given as input, with or without -s, the output to terminal is an error message. If both a valid file–or files–are input, but so is an invalid not-file and/or file that doesn't exist, then the valid files will be deleted, but an error message will still display.

I have no idea why one sentence is expected here, or how three bullet points are expected to specifically get divvied up into four paragraphs–where one paragraph is one sentence?? Give me a break here.

the < if [$# -eq 0] ; then > block checks if there were arguments passed when the script was called. If no arguments were passed, it outputs a usage message and exits the script with the 1 error code. the $# variable is used to count the number of arguments passed to the script.

the < if ["$1" = "-s"] ; then block is used if the first block is untrue to check if the first argument passed to the script is -s. If it is, then the shift command is used to remove the -s argument, and if not, the date, username, and file name (or, arguments passed to the program–not including a -s) are written to the file. Then, either way, the $@ variable is used to remove any files listed as arguments passed to the program. Then it exits with code 0.

In the space below, cut and paste and provide all execution tests performed on this script. This should include at least 10 different tests.

```
cari@Thinky:~$ touch a.txt b.txt c.txt
cari@Thinky:~$ ./part1 a.txt
bash: ./part1: No such file or directory
cari@Thinky:~$ ls
 a.txt                                   'Lab 1 Materials.zip'
 b.txt                                    Lab2.X.zip
 CST204                                   logfile
 CST240                                   MPLABXProjects
'CST240 Lab4 - Files-Directories, Bash Encryption.docx'   Music
 c.txt                                    Pictures
 Desktop                                  Public
 Documents                                Templates
 Downloads                                Videos
cari@Thinky:~$ rm a.txt
cari@Thinky:~$ rm b.txt
cari@Thinky:~$ rm c.txt
cari@Thinky:~$ ls
```

```
 CST204                             logfile
 CST240                             MPLABXProjects
'CST240 Lab4 - Files-Directories, Bash Encryption.docx'   Music
 Desktop                            Pictures
 Documents                          Public
 Downloads                          Templates
'Lab 1 Materials.zip'                      Videos
 Lab2.X.zip
cari@Thinky:~$ rm 'CST2
CST204/
CST240/
CST240 Lab4 - Files-Directories, Bash Encryption.docx
cari@Thinky:~$ cd CST240
cari@Thinky:~/CST240$ ls
HW1  LAB  Quizzes
cari@Thinky:~/CST240$ cd LAB
cari@Thinky:~/CST240/LAB$ cd L4
cari@Thinky:~/CST240/LAB/L4$ ls
d.txt  part1  part1.sh
cari@Thinky:~/CST240/LAB/L4$ touch a.txt b.txt c.txt
cari@Thinky:~/CST240/LAB/L4$ ls
a.txt  b.txt  c.txt  d.txt  part1  part1.sh
cari@Thinky:~/CST240/LAB/L4$ rm part1
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh a.txt
bash: ./part1.sh: Permission denied
cari@Thinky:~/CST240/LAB/L4$ ls
a.txt  b.txt  c.txt  d.txt  part1.sh
cari@Thinky:~/CST240/LAB/L4$ chmod 777 part1.sh
cari@Thinky:~/CST240/LAB/L4$ ls
a.txt  b.txt  c.txt  d.txt  part1.sh
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh a.txt
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh b.txt
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh c.txt
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh -s d.txt
cari@Thinky:~/CST240/LAB/L4$ ls
part1.sh
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh
Usage: ./part1.sh [-s] list of files or directories
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh a
```

```
/bin/rm: cannot remove 'a': No such file or directory
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh -s a
/bin/rm: cannot remove 'a': No such file or directory
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh a.txt
/bin/rm: cannot remove 'a.txt': No such file or directory
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh -s a.txt
/bin/rm: cannot remove 'a.txt': No such file or directory
cari@Thinky:~/CST240/LAB/L4$ touch a.txt b.txt c.txt d.txt
cari@Thinky:~/CST240/LAB/L4$ ./part.sh -s a.txt b.txt c.txt d.txt
bash: ./part.sh: No such file or directory
cari@Thinky:~/CST240/LAB/L4$ ls
a.txt  b.txt  c.txt  d.txt  part1.sh
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh -s a.txt b.txt c.txt d.txt
cari@Thinky:~/CST240/LAB/L4$ ls
part1.sh
cari@Thinky:~/CST240/LAB/L4$ touch a.txt b.txt c.txt d.txt
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh -s a.txt a
/bin/rm: cannot remove 'a': No such file or directory
cari@Thinky:~/CST240/LAB/L4$ ls
b.txt  c.txt  d.txt  part1.sh
cari@Thinky:~/CST240/LAB/L4$ ./part1.sh -s b b.txt
/bin/rm: cannot remove 'b': No such file or directory
cari@Thinky:~/CST240/LAB/L4$ ls
c.txt  d.txt  part1.sh
```

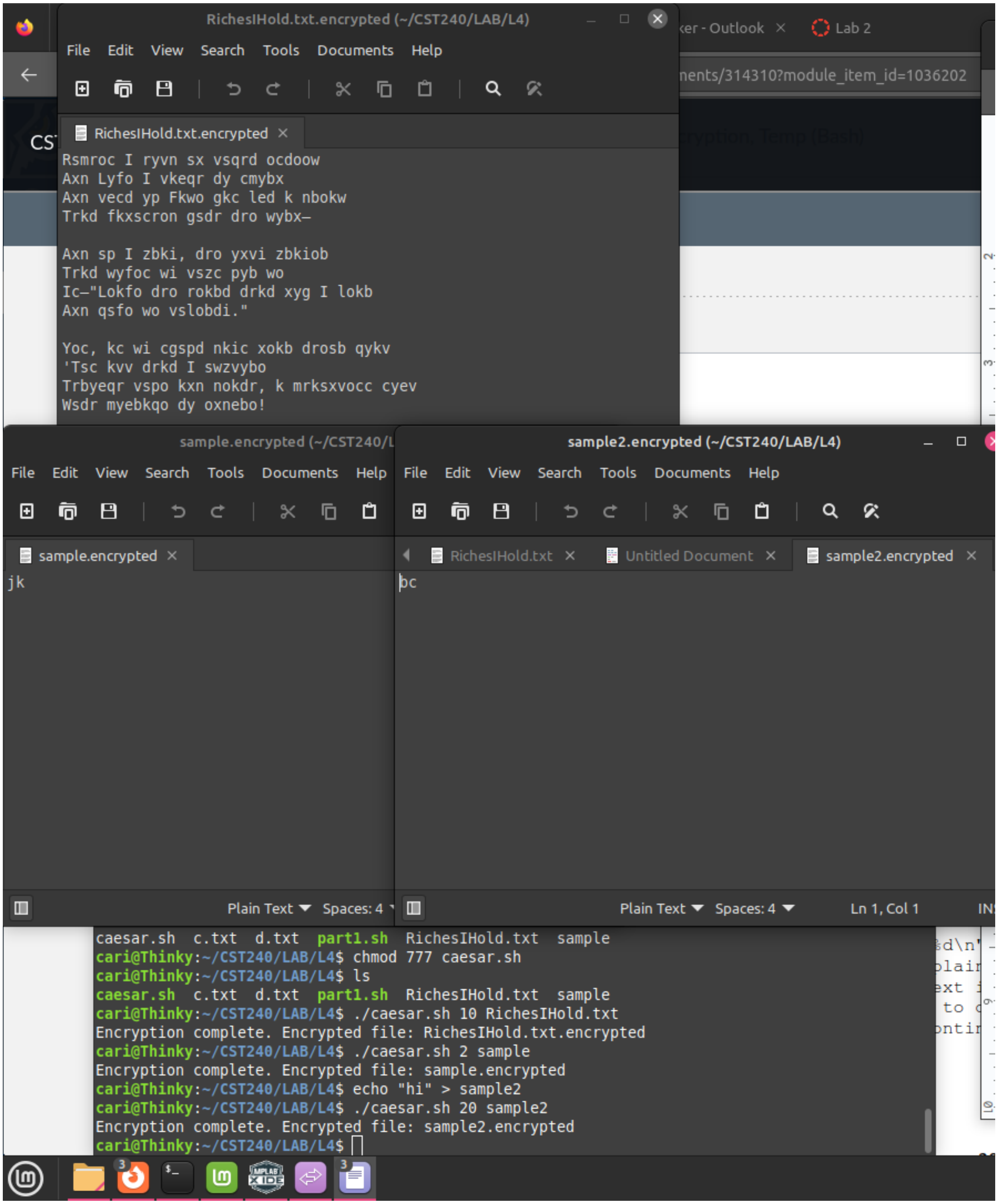
Grading:
Description: 5 pts
Tests 15 pts

2. Write a Bash script named caesar.sh that implements Caesar encryption taking in 2 parameters: <key> <filename>.

Accept only lower-case characters a-z: ignore all other characters. Test it out.

Use a conversion algorithm method as used in Lab 2.

Show in the space below your commands and output required to encrypt “RichesIHold.txt” with a key of 10. Your script should produce something like the following (except for the “RichesIHold.txt” file):
Copy and paste or show a screenshot of the result of encrypting “RichesIHold.txt” using your bash script with a key of 10 in space below:


RichesIHold.txt.encrypted (~/CST240/LAB/L4)
File Edit View Search Tools Documents Help
RichesIHold.txt.encrypted
sample.encrypted (~/CST240/L
sample.encrypted
sample2.encrypted (~/CST240/LAB/L4)
RichesIHold.txt
Untitled Document
sample2.encrypted
Plain Text
Spaces: 4
Ln 1, Col 1

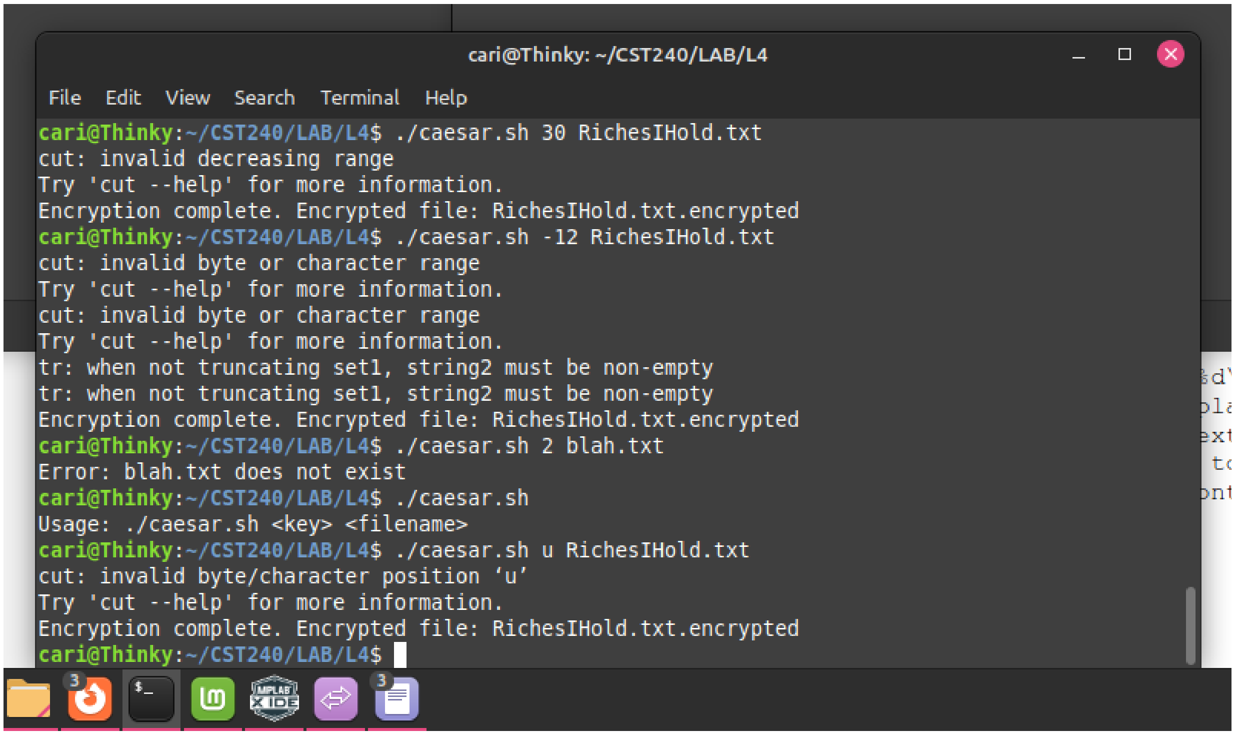
```
Rsmroc I ryvn sx vsqrd ocdoow
Axn Lyfo I vkeqr dy cmybx
Axn vecd yp Fkwo gkc led k nbokw
Trkd fkxscron gsdr dro wybx—

Axn sp I zbki, dro yxvi zbkiob
Trkd wyfoc wi vszc pyb wo
Ic—"Lokfo dro rokbd drkd xyg I lokb
Axn qsfo wo vslobdi."

Yoc, kc wi cgspd nkic xokb drosb qykv
'Tsc kvv drkd I swzvybo
Trbyeqr vspo kxn nokdr, k mrksxvocc cyev
Wsdr myebkqo dy oxnebo!
```

```
jk
```

```
bc
```

```
caesar.sh  c.txt  d.txt  part1.sh  RichesIHold.txt  sample
cari@Thinky:~/CST240/LAB/L4$ chmod 777 caesar.sh
cari@Thinky:~/CST240/LAB/L4$ ls
caesar.sh  c.txt  d.txt  part1.sh  RichesIHold.txt  sample
cari@Thinky:~/CST240/LAB/L4$ ./caesar.sh 10 RichesIHold.txt
Encryption complete. Encrypted file: RichesIHold.txt.encrypted
cari@Thinky:~/CST240/LAB/L4$ ./caesar.sh 2 sample
Encryption complete. Encrypted file: sample.encrypted
cari@Thinky:~/CST240/LAB/L4$ echo "hi" > sample2
cari@Thinky:~/CST240/LAB/L4$ ./caesar.sh 20 sample2
Encryption complete. Encrypted file: sample2.encrypted
cari@Thinky:~/CST240/LAB/L4$
```

Show other tests of this script including boundary conditions and unexpected data below:


cari@Thinky: ~/CST240/LAB/L4
File Edit View Search Terminal Help

```
cari@Thinky:~/CST240/LAB/L4$ ./caesar.sh 30 RichesIHold.txt
cut: invalid decreasing range
Try 'cut --help' for more information.
Encryption complete. Encrypted file: RichesIHold.txt.encrypted
cari@Thinky:~/CST240/LAB/L4$ ./caesar.sh -12 RichesIHold.txt
cut: invalid byte or character range
Try 'cut --help' for more information.
cut: invalid byte or character range
Try 'cut --help' for more information.
tr: when not truncating set1, string2 must be non-empty
tr: when not truncating set1, string2 must be non-empty
Encryption complete. Encrypted file: RichesIHold.txt.encrypted
cari@Thinky:~/CST240/LAB/L4$ ./caesar.sh 2 blah.txt
Error: blah.txt does not exist
cari@Thinky:~/CST240/LAB/L4$ ./caesar.sh
Usage: ./caesar.sh <key> <filename>
cari@Thinky:~/CST240/LAB/L4$ ./caesar.sh u RichesIHold.txt
cut: invalid byte/character position ‘u’
Try 'cut --help' for more information.
Encryption complete. Encrypted file: RichesIHold.txt.encrypted
cari@Thinky:~/CST240/LAB/L4$ 
```

3. Write a Bash script named railfence.sh that implements Rail Fence encryption taking in 2 parameters: <key> <filename>, similar to the Caesar script in part 1.

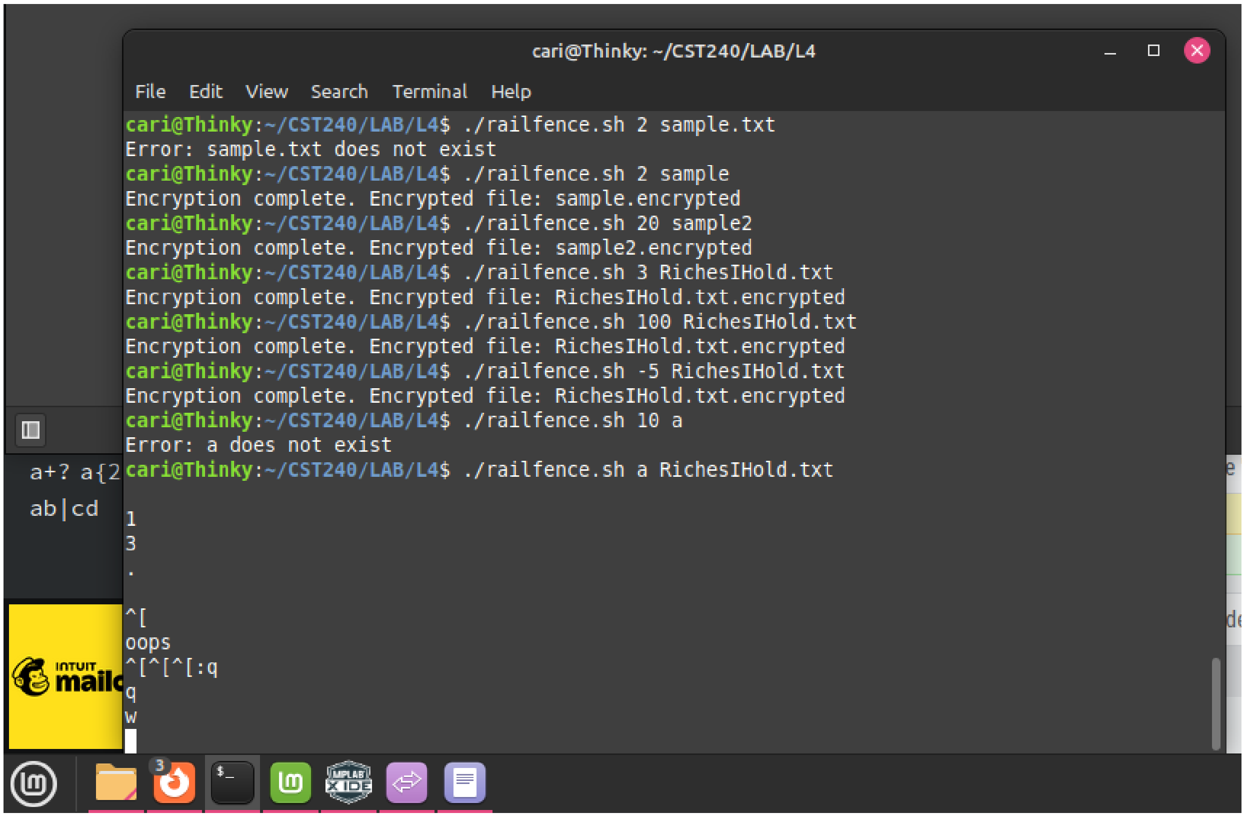
Accept only lower-case characters a-z: ignore all other characters. Test it out.

Grading:

Code working correctly:	50 pts
Code documentation:	10 pts
Code testing:	20 pts

For documentation, be sure to include your name, the date, the class and assignment at the top.
Also be sure to describe what every major section of code is doing in a comment.

When testing, be sure to test boundary conditions and unexpected data along with expected data.


cari@Thinky: ~/CST240/LAB/L4
File Edit View Search Terminal Help

```
cari@Thinky:~/CST240/LAB/L4$ ./railfence.sh 2 sample.txt
Error: sample.txt does not exist
cari@Thinky:~/CST240/LAB/L4$ ./railfence.sh 2 sample
Encryption complete. Encrypted file: sample.encrypted
cari@Thinky:~/CST240/LAB/L4$ ./railfence.sh 20 sample2
Encryption complete. Encrypted file: sample2.encrypted
cari@Thinky:~/CST240/LAB/L4$ ./railfence.sh 3 RichesIHold.txt
Encryption complete. Encrypted file: RichesIHold.txt.encrypted
cari@Thinky:~/CST240/LAB/L4$ ./railfence.sh 100 RichesIHold.txt
Encryption complete. Encrypted file: RichesIHold.txt.encrypted
cari@Thinky:~/CST240/LAB/L4$ ./railfence.sh -5 RichesIHold.txt
Encryption complete. Encrypted file: RichesIHold.txt.encrypted
cari@Thinky:~/CST240/LAB/L4$ ./railfence.sh 10 a
Error: a does not exist
cari@Thinky:~/CST240/LAB/L4$ ./railfence.sh a RichesIHold.txt

1
3
.

^[
oops
^[^[^[:q
q
w
```