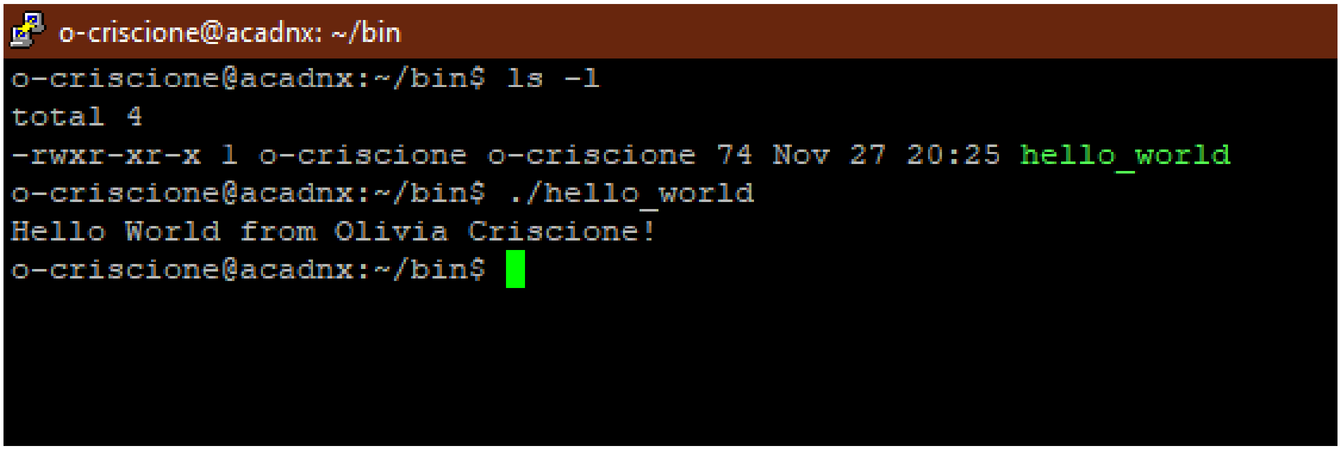
Olivia Criscione CISS100 Final Project

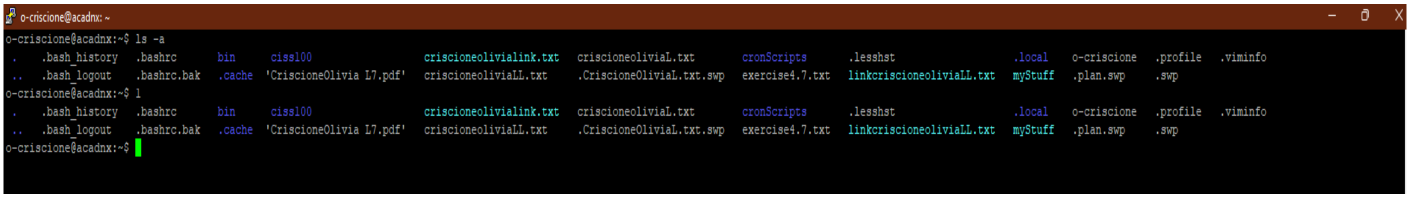
Exercise 1: Writing Your First Script and Getting It to Work



```
o-criscione@acadnx:~/bin$ ls -l
total 4
-rwxr-xr-x 1 o-criscione o-criscione 74 Nov 27 20:25 hello_world
o-criscione@acadnx:~/bin$ ./hello_world
Hello World from Olivia Criscione!
o-criscione@acadnx:~/bin$
```

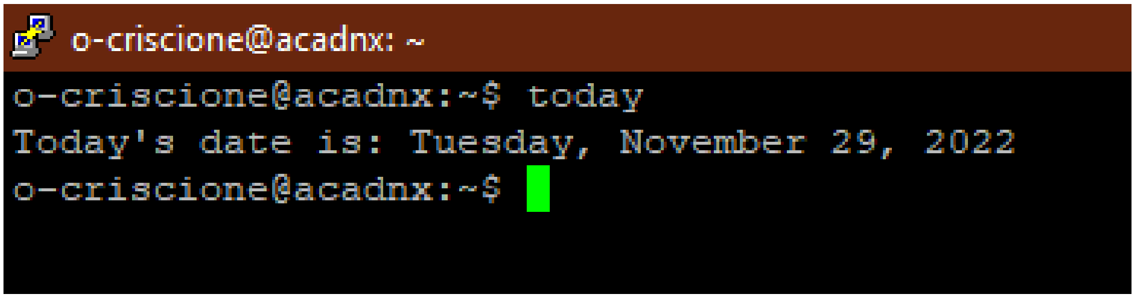
Exercise 2: Editing The Scripts You Already Have

- Part A+B:



```
o-criscione@acadnx:~$ ls -a
.    .bash_history   .bashrc      bin      ciss100                    criscioneolivialink.txt   criscioneoliviaL.txt        cronScripts       .lesshst                    .local    o-criscione   .profile   .viminfo
..   .bash_logout    .bashrc.bak  .cache   'CriscioneOlivia L7.pdf'   criscioneoliviaLL.txt     .CriscioneOliviaL.txt.swp   exercise4.7.txt   linkcriscioneoliviaLL.txt   myStuff   .plan.swp     .swp
o-criscione@acadnx:~$ l
.    .bash_history   .bashrc      bin      ciss100                    criscioneolivialink.txt   criscioneoliviaL.txt        cronScripts       .lesshst                    .local    o-criscione   .profile   .viminfo
..   .bash_logout    .bashrc.bak  .cache   'CriscioneOlivia L7.pdf'   criscioneoliviaLL.txt     .CriscioneOliviaL.txt.swp   exercise4.7.txt   linkcriscioneoliviaLL.txt   myStuff   .plan.swp     .swp
o-criscione@acadnx:~$
```

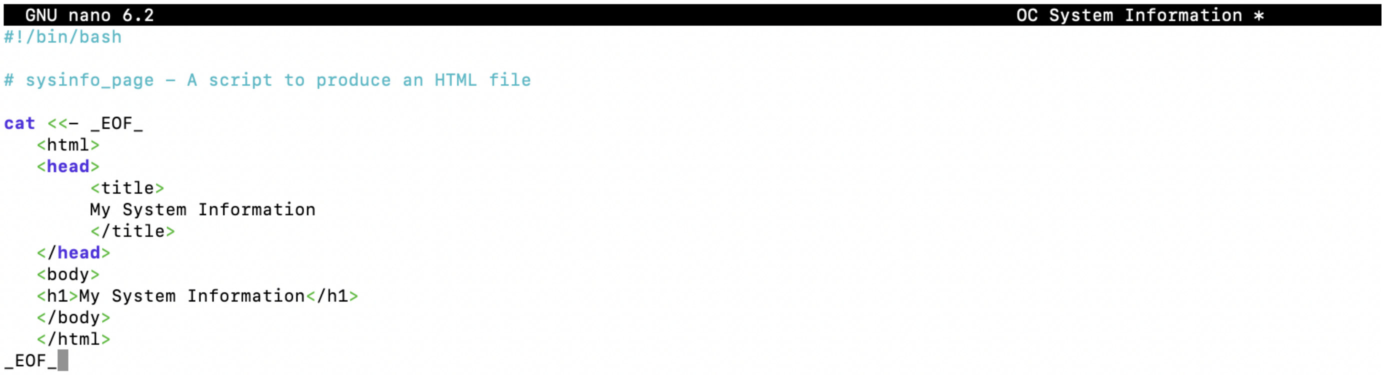
- Part C:



```
o-criscione@acadnx:~$ today
Today's date is: Tuesday, November 29, 2022
o-criscione@acadnx:~$
```

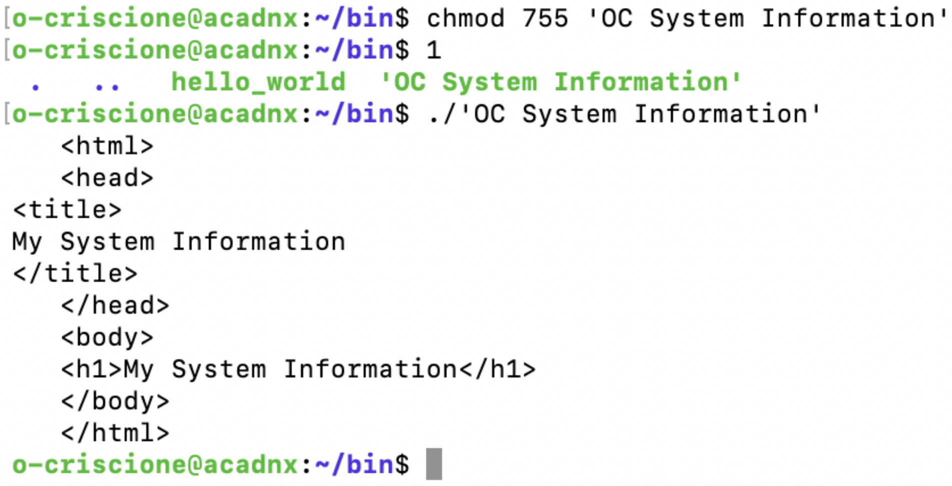
Exercise 3: Here Scripts

- Part A:



```
  GNU nano 6.2                                                                                     OC System Information *
#!/bin/bash

# sysinfo_page - A script to produce an HTML file

cat <<- _EOF_
   <html>
   <head>
        <title>
        My System Information
        </title>
   </head>
   <body>
   <h1>My System Information</h1>
   </body>
   </html>
_EOF_
```

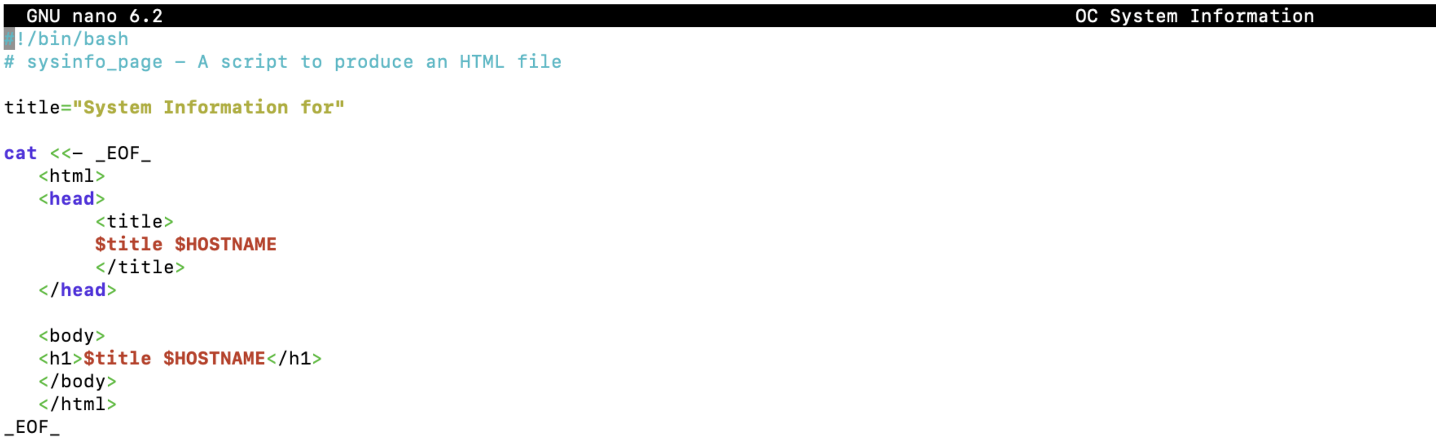
- Part B:



```
[o-criscione@acadnx:~/bin$ chmod 755 'OC System Information'
[o-criscione@acadnx:~/bin$ 1
 .   ..   hello_world  'OC System Information'
[o-criscione@acadnx:~/bin$ ./'OC System Information'
   <html>
   <head>
<title>
My System Information
</title>
   </head>
   <body>
   <h1>My System Information</h1>
   </body>
   </html>
o-criscione@acadnx:~/bin$ 
```

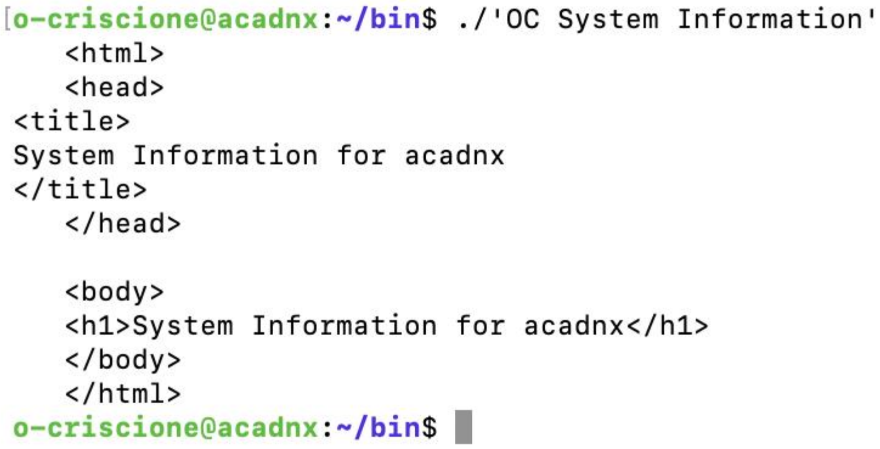
Exercise 4: Variables

- Part A:



```
#!/bin/bash
# sysinfo_page - A script to produce an HTML file

title="System Information for"

cat <<- _EOF_
   <html>
   <head>
        <title>
        $title $HOSTNAME
        </title>
   </head>

   <body>
   <h1>$title $HOSTNAME</h1>
   </body>
   </html>
_EOF_
```

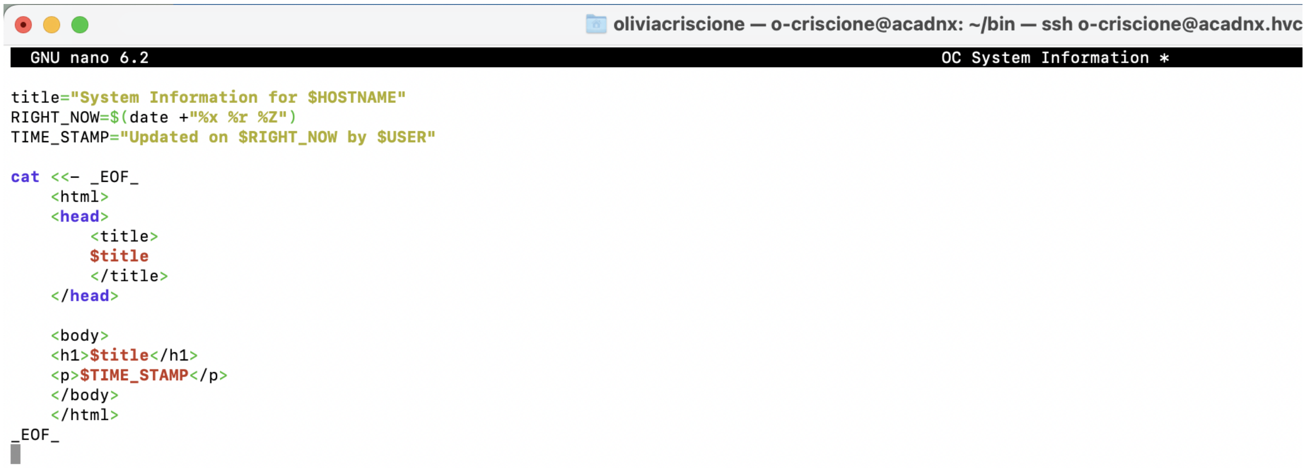
- Part B:



```
[o-criscione@acadnx:~/bin$ ./'OC System Information'
    <html>
    <head>
<title>
System Information for acadnx
</title>
    </head>

    <body>
    <h1>System Information for acadnx</h1>
    </body>
    </html>
o-criscione@acadnx:~/bin$ 
```

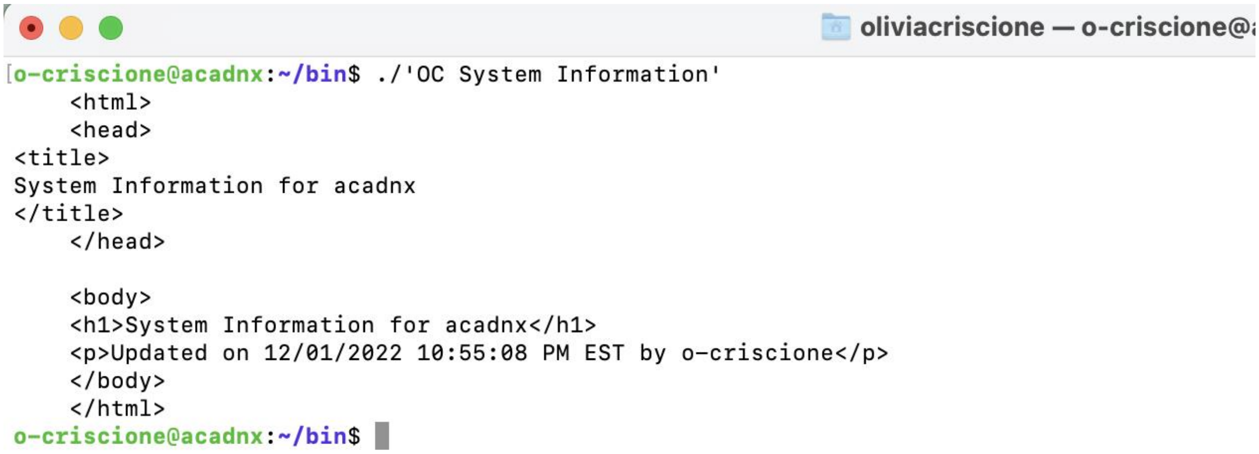
Exercise 5: Command Substitution and Constants

- Part A:


oliviacriscione — o-criscione@acadnx: ~/bin — ssh o-criscione@acadnx.hvc
GNU nano 6.2
OC System Information *

```
title="System Information for $HOSTNAME"
RIGHT_NOW=$(date +"%x %r %Z")
TIME_STAMP="Updated on $RIGHT_NOW by $USER"

cat <<- _EOF_
    <html>
    <head>
        <title>
        $title
        </title>
    </head>

    <body>
    <h1>$title</h1>
    <p>$TIME_STAMP</p>
    </body>
    </html>
_EOF_
```

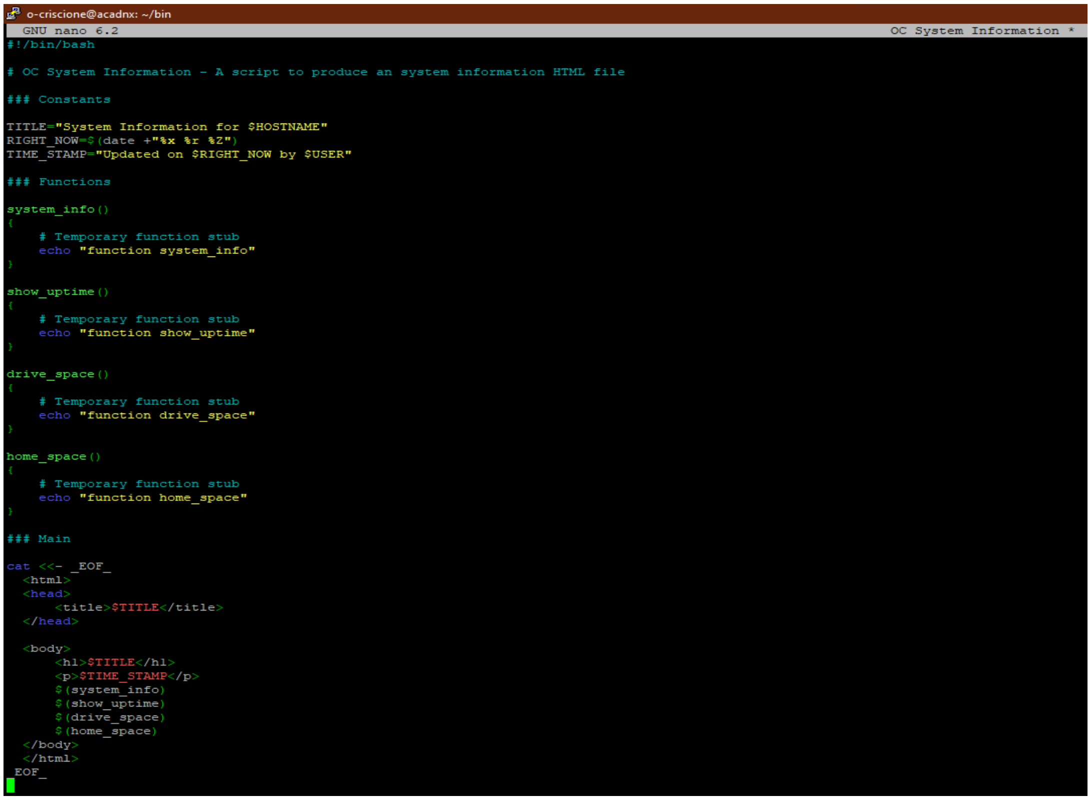
- Part B:



```
[o-criscione@acadnx:~/bin$ ./'OC System Information'
    <html>
    <head>
<title>
System Information for acadnx
</title>
    </head>

    <body>
    <h1>System Information for acadnx</h1>
    <p>Updated on 12/01/2022 10:55:08 PM EST by o-criscione</p>
    </body>
    </html>
o-criscione@acadnx:~/bin$
```

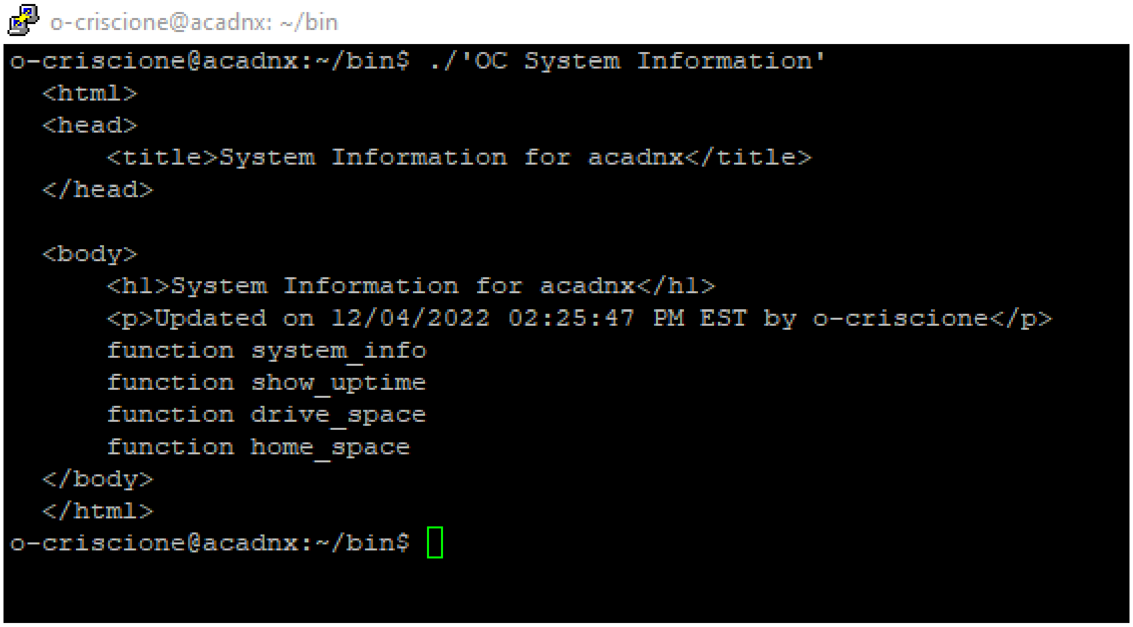
Exercise 6 Shell Functions:

- Part A:



```
o-criscione@acadnx: ~/bin
  GNU nano 6.2                                                                              OC System Information *
#!/bin/bash

# OC System Information - A script to produce an system information HTML file

### Constants

TITLE="System Information for $HOSTNAME"
RIGHT_NOW=$(date +"%x %r %Z")
TIME_STAMP="Updated on $RIGHT_NOW by $USER"

### Functions

system_info()
{
    # Temporary function stub
    echo "function system_info"
}

show_uptime()
{
    # Temporary function stub
    echo "function show_uptime"
}

drive_space()
{
    # Temporary function stub
    echo "function drive_space"
}

home_space()
{
    # Temporary function stub
    echo "function home_space"
}

### Main

cat <<- _EOF_
  <html>
  <head>
      <title>$TITLE</title>
  </head>

  <body>
      <h1>$TITLE</h1>
      <p>$TIME_STAMP</p>
      $(system_info)
      $(show_uptime)
      $(drive_space)
      $(home_space)
  </body>
  </html>
 EOF_
```

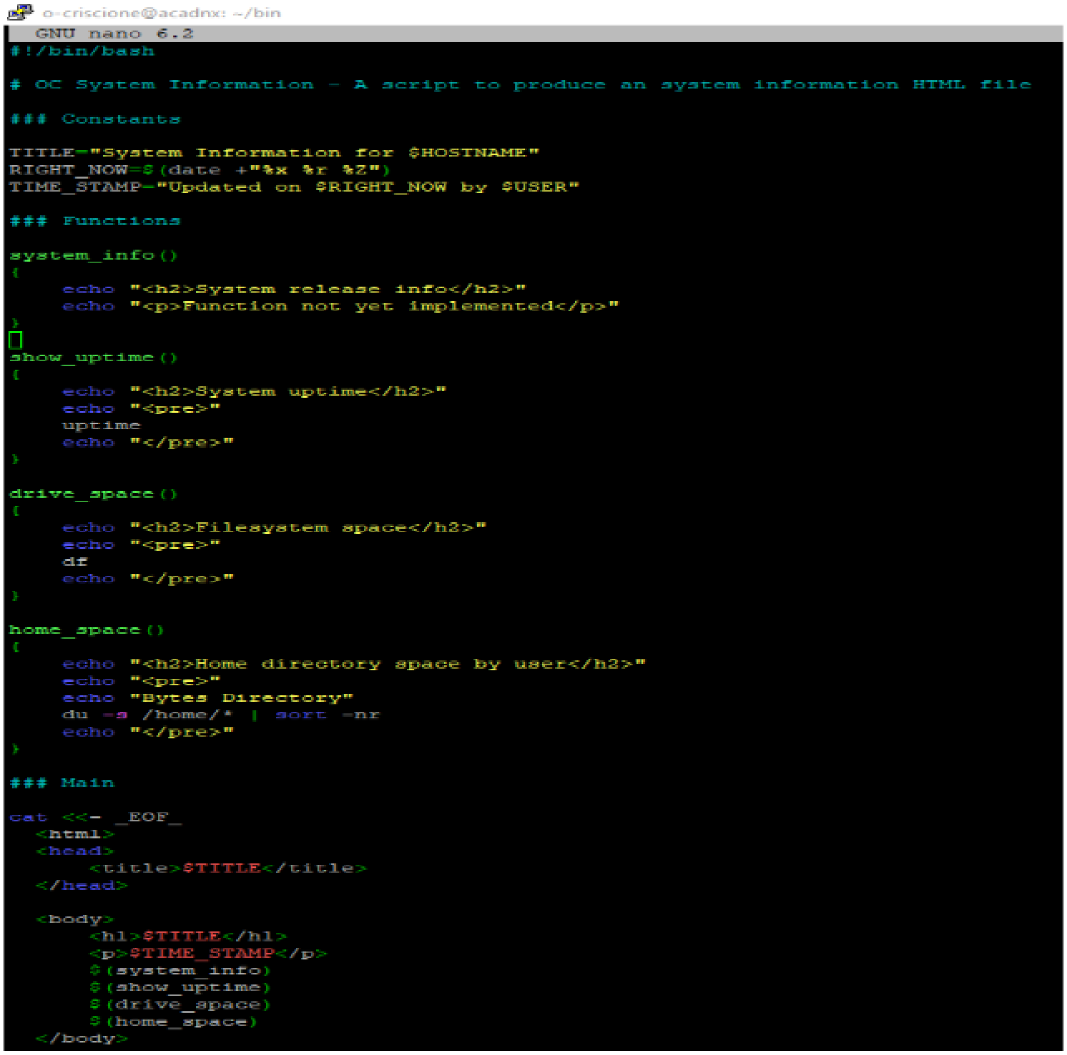
- Part B:



```
o-criscione@acadnx: ~/bin
o-criscione@acadnx:~/bin$ ./'OC System Information'
  <html>
  <head>
      <title>System Information for acadnx</title>
  </head>

  <body>
      <h1>System Information for acadnx</h1>
      <p>Updated on 12/04/2022 02:25:47 PM EST by o-criscione</p>
      function system_info
      function show_uptime
      function drive_space
      function home_space
  </body>
  </html>
o-criscione@acadnx:~/bin$
```

Exercise 7 Some Real Work:

- Part A:

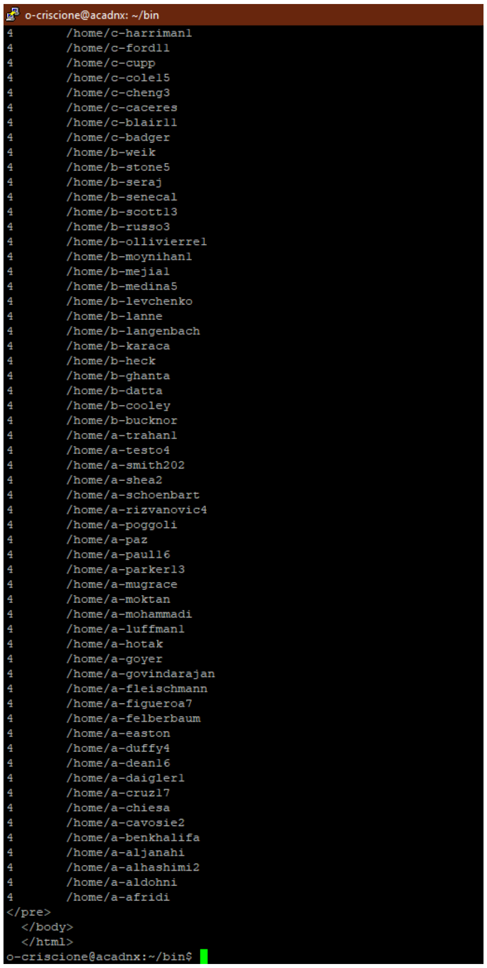


```
#!/bin/bash

# OC System Information - A script to produce an system information HTML file

### Constants

TITLE="System Information for $HOSTNAME"
RIGHT_NOW=$(date +"%x %r %Z")
TIME_STAMP="Updated on $RIGHT_NOW by $USER"

### Functions

system_info()
{
     echo "<h2>System release info</h2>"
     echo "<p>Function not yet implemented</p>"
}

show_uptime()
{
     echo "<h2>System uptime</h2>"
     echo "<pre>"
     uptime
     echo "</pre>"
}

drive_space()
{
     echo "<h2>Filesystem space</h2>"
     echo "<pre>"
     df
     echo "</pre>"
}

home_space()
{
     echo "<h2>Home directory space by user</h2>"
     echo "<pre>"
     echo "Bytes Directory"
     du -s /home/* | sort -nr
     echo "</pre>"
}

### Main

cat <<- _EOF_
  <html>
  <head>
      <title>$TITLE</title>
  </head>

  <body>
      <h1>$TITLE</h1>
      <p>$TIME_STAMP</p>
      $(system_info)
      $(show_uptime)
      $(drive_space)
      $(home_space)
  </body>
```

(Part B on next page)

Exercise 7 Continued:

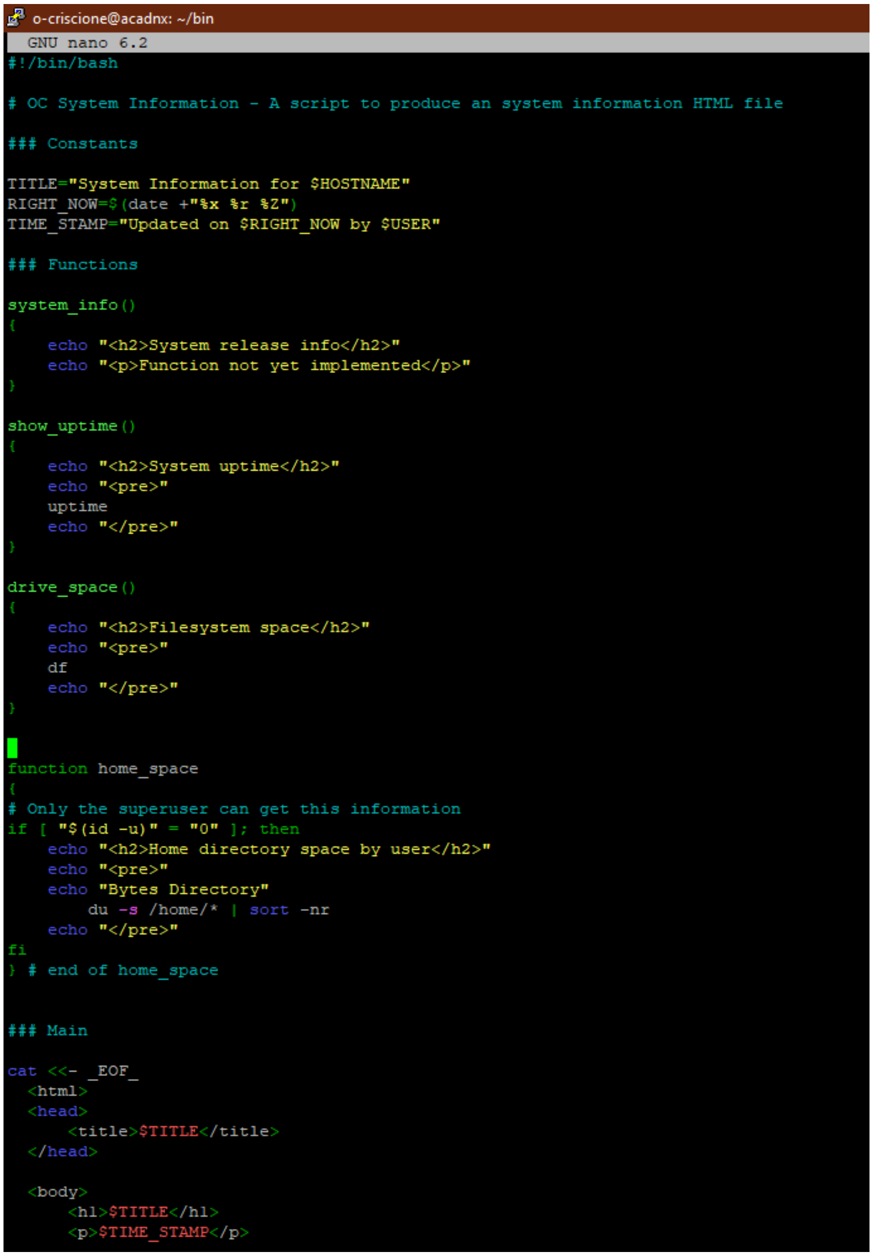
- Part B:



```
4       /home/c-harrimanl
4       /home/c-fordll
4       /home/c-cupp
4       /home/c-cole15
4       /home/c-cheng3
4       /home/c-caceres
4       /home/c-blairll
4       /home/c-badger
4       /home/b-weik
4       /home/b-stone5
4       /home/b-seraj
4       /home/b-senecal
4       /home/b-scott13
4       /home/b-russo3
4       /home/b-ollivierrel
4       /home/b-moynihanl
4       /home/b-mejial
4       /home/b-medina5
4       /home/b-levchenko
4       /home/b-lanne
4       /home/b-langenbach
4       /home/b-karaca
4       /home/b-heck
4       /home/b-ghanta
4       /home/b-datta
4       /home/b-cooley
4       /home/b-bucknor
4       /home/a-trahanl
4       /home/a-testo4
4       /home/a-smith202
4       /home/a-shea2
4       /home/a-schoenbart
4       /home/a-rizvanovic4
4       /home/a-poggoli
4       /home/a-paz
4       /home/a-paul16
4       /home/a-parker13
4       /home/a-mugrace
4       /home/a-moktan
4       /home/a-mohammadi
4       /home/a-luffmanl
4       /home/a-hotak
4       /home/a-goyer
4       /home/a-govindarajan
4       /home/a-fleischmann
4       /home/a-figueroa7
4       /home/a-felberbaum
4       /home/a-easton
4       /home/a-duffy4
4       /home/a-dean16
4       /home/a-daiglerl
4       /home/a-cruz17
4       /home/a-chiesa
4       /home/a-cavosie2
4       /home/a-benkhalifa
4       /home/a-aljanahi
4       /home/a-alhashimi2
4       /home/a-aldohni
4       /home/a-afridi
</pre>
  </body>
  </html>
o-criscione@acadnx:~/bin$
```

Exercise 8 Flow Control – Part 1:

- Part A:

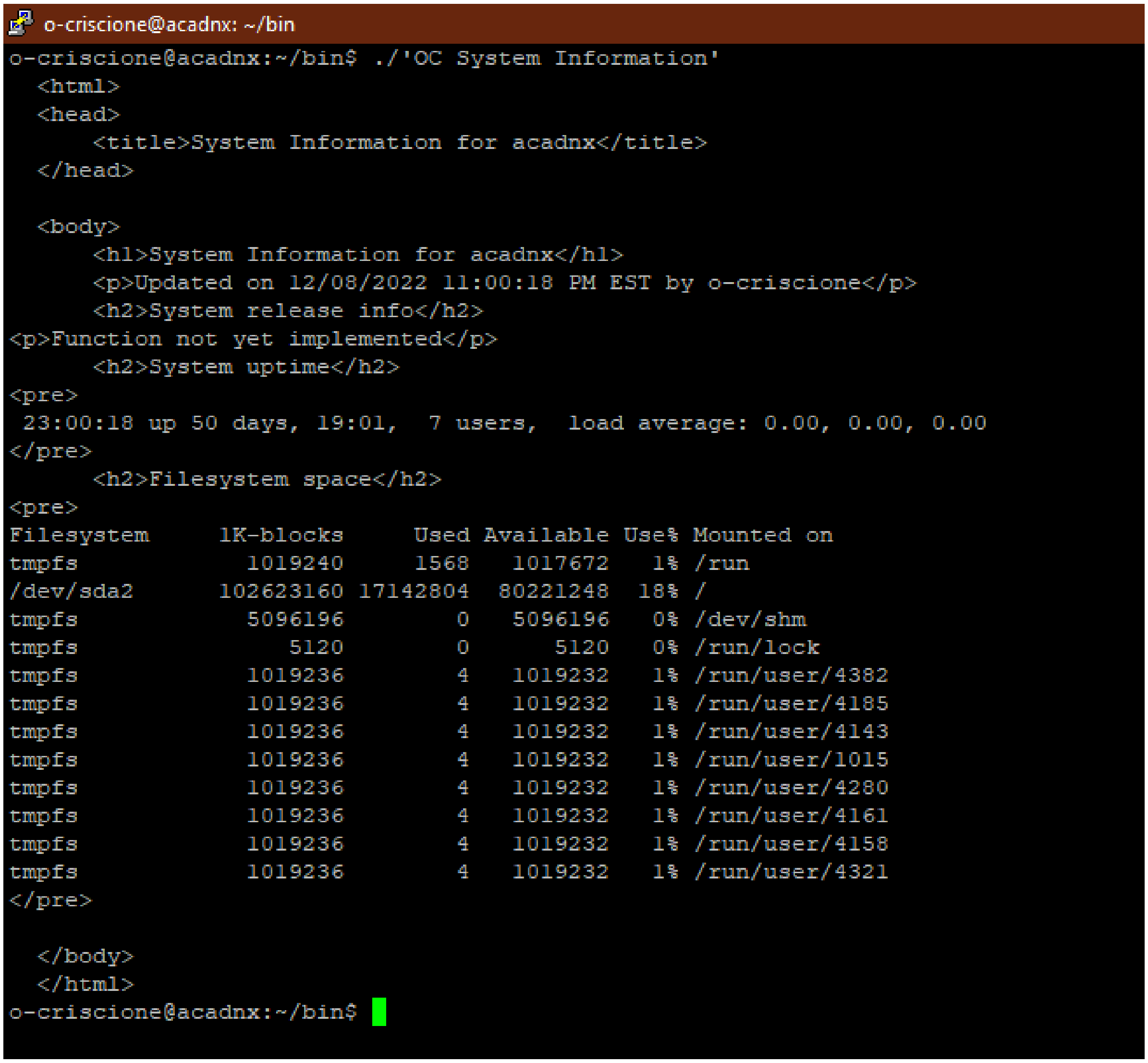


```
#!/bin/bash

# OC System Information - A script to produce an system information HTML file

### Constants

TITLE="System Information for $HOSTNAME"
RIGHT_NOW=$(date +"%x %r %Z")
TIME_STAMP="Updated on $RIGHT_NOW by $USER"

### Functions

system_info()
{
    echo "<h2>System release info</h2>"
    echo "<p>Function not yet implemented</p>"
}

show_uptime()
{
    echo "<h2>System uptime</h2>"
    echo "<pre>"
    uptime
    echo "</pre>"
}

drive_space()
{
    echo "<h2>Filesystem space</h2>"
    echo "<pre>"
    df
    echo "</pre>"
}


function home_space
{
# Only the superuser can get this information
if [ "$(id -u)" = "0" ]; then
    echo "<h2>Home directory space by user</h2>"
    echo "<pre>"
    echo "Bytes Directory"
        du -s /home/* | sort -nr
    echo "</pre>"
fi
} # end of home_space


### Main

cat <<- _EOF_
  <html>
  <head>
      <title>$TITLE</title>
  </head>

  <body>
      <h1>$TITLE</h1>
      <p>$TIME_STAMP</p>
```

- Part B on next page

Exercise 8 continued:

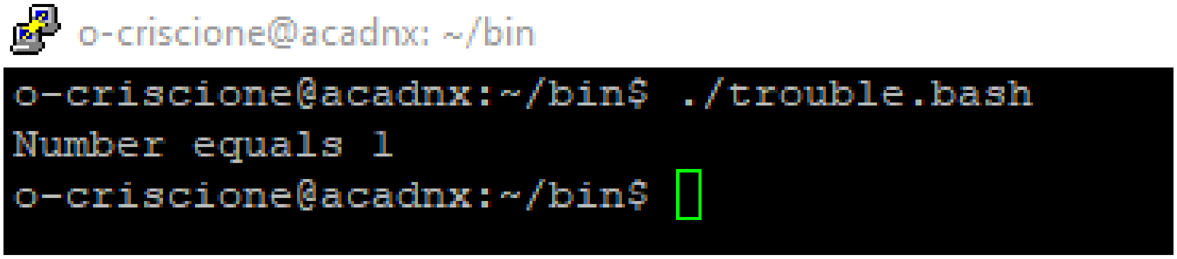
- Part B:



```
o-criscione@acadnx:~/bin$ ./'OC System Information'
  <html>
  <head>
      <title>System Information for acadnx</title>
  </head>

  <body>
      <h1>System Information for acadnx</h1>
      <p>Updated on 12/08/2022 11:00:18 PM EST by o-criscione</p>
      <h2>System release info</h2>
<p>Function not yet implemented</p>
      <h2>System uptime</h2>
<pre>
 23:00:18 up 50 days, 19:01,  7 users,  load average: 0.00, 0.00, 0.00
</pre>
      <h2>Filesystem space</h2>
<pre>
Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs            1019240     1568   1017672   1% /run
/dev/sda2      102623160 17142804  80221248  18% /
tmpfs            5096196        0   5096196   0% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs            1019236        4   1019232   1% /run/user/4382
tmpfs            1019236        4   1019232   1% /run/user/4185
tmpfs            1019236        4   1019232   1% /run/user/4143
tmpfs            1019236        4   1019232   1% /run/user/1015
tmpfs            1019236        4   1019232   1% /run/user/4280
tmpfs            1019236        4   1019232   1% /run/user/4161
tmpfs            1019236        4   1019232   1% /run/user/4158
tmpfs            1019236        4   1019232   1% /run/user/4321
</pre>

  </body>
  </html>
o-criscione@acadnx:~/bin$
```

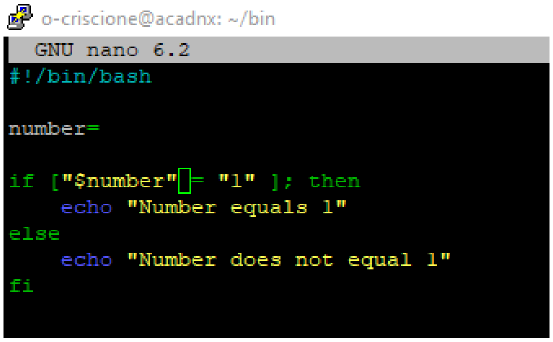
Exercise 9:

- Part A:



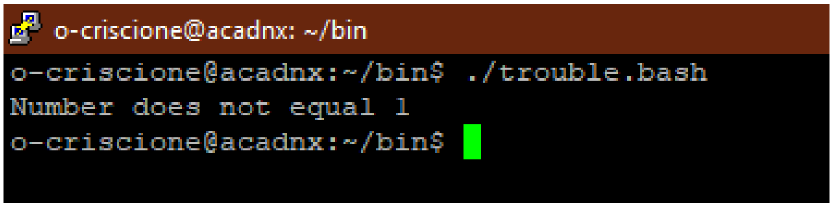
```
o-criscione@acadnx:~/bin$ ./trouble.bash
Number equals 1
o-criscione@acadnx:~/bin$
```

- Part B SS1:



```
#!/bin/bash

number=

if ["$number" = "1" ]; then
    echo "Number equals 1"
else
    echo "Number does not equal 1"
fi
```

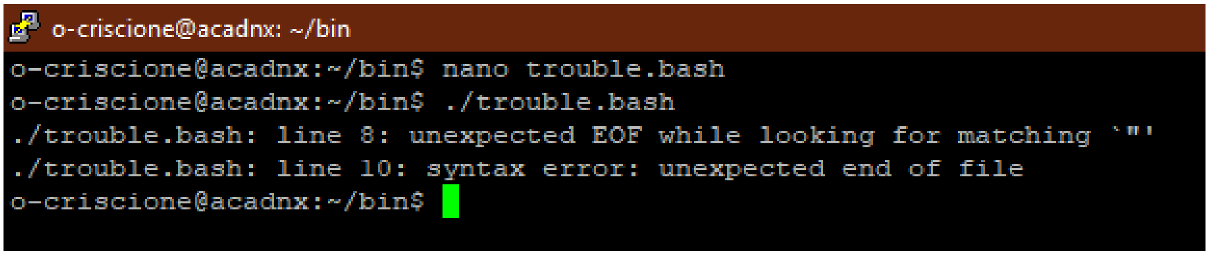
- Part B SS2:



```
o-criscione@acadnx:~/bin$ ./trouble.bash
Number does not equal 1
o-criscione@acadnx:~/bin$
```

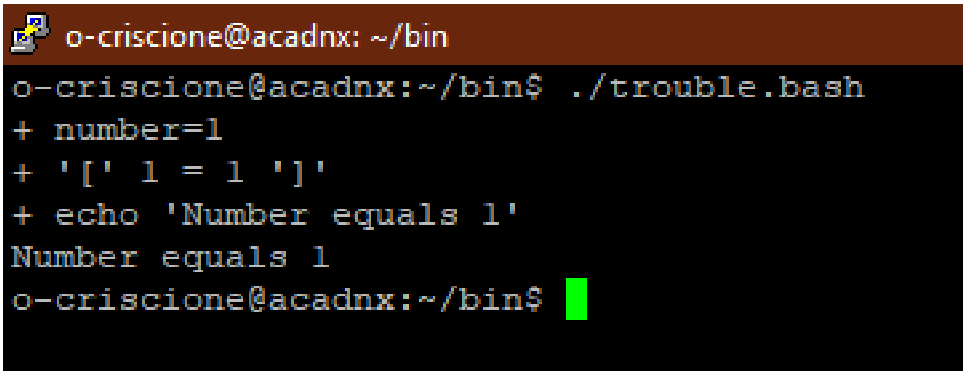
Exercise 9:

- Part C:



```
o-criscione@acadnx:~/bin$ nano trouble.bash
o-criscione@acadnx:~/bin$ ./trouble.bash
./trouble.bash: line 8: unexpected EOF while looking for matching `"'
./trouble.bash: line 10: syntax error: unexpected end of file
o-criscione@acadnx:~/bin$
```

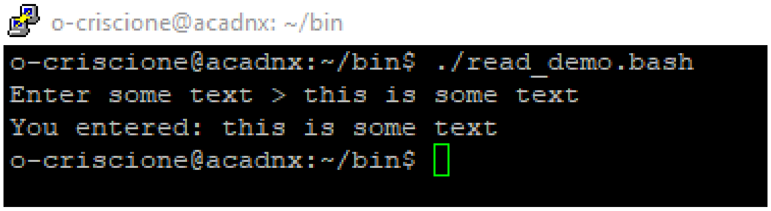
- Part D:



```
o-criscione@acadnx:~/bin$ ./trouble.bash
+ number=1
+ '[' 1 = 1 ']'
+ echo 'Number equals 1'
Number equals 1
o-criscione@acadnx:~/bin$
```

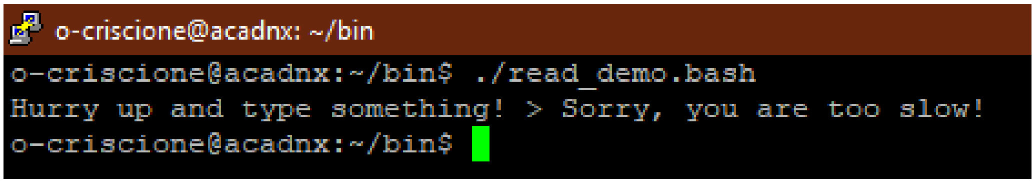
Exercise 10 Keyboard Input and Arithmetic:

- Part A:



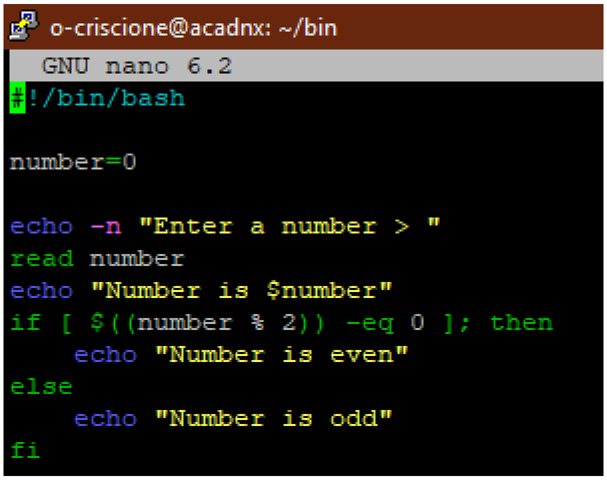
```
o-criscione@acadnx:~/bin$ ./read_demo.bash
Enter some text > this is some text
You entered: this is some text
o-criscione@acadnx:~/bin$
```

- Part B:



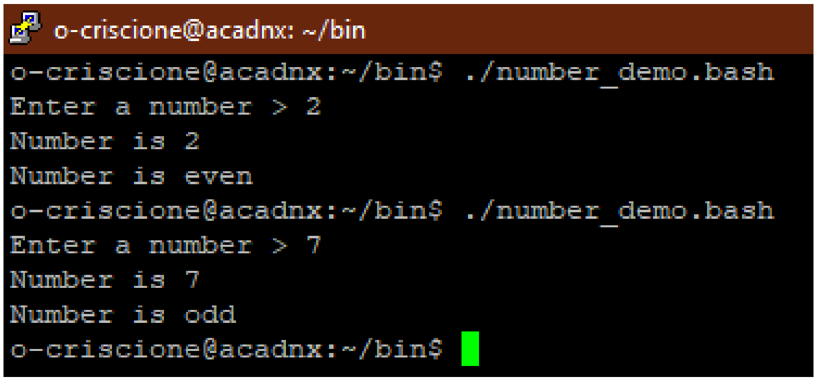
```
o-criscione@acadnx:~/bin$ ./read_demo.bash
Hurry up and type something! > Sorry, you are too slow!
o-criscione@acadnx:~/bin$
```

- Part C SS1:



```
o-criscione@acadnx: ~/bin
  GNU nano 6.2
#!/bin/bash

number=0

echo -n "Enter a number > "
read number
echo "Number is $number"
if [ $((number % 2)) -eq 0 ]; then
    echo "Number is even"
else
    echo "Number is odd"
fi
```

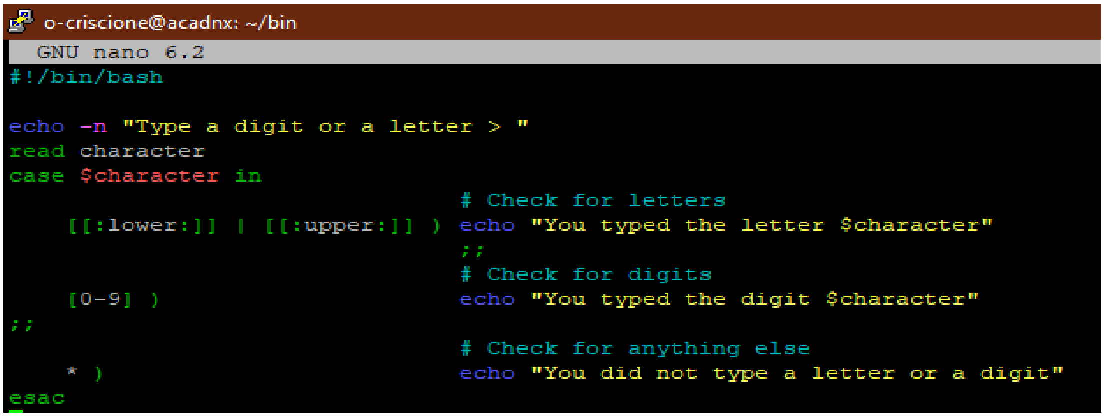
- Part C SS2:



```
o-criscione@acadnx: ~/bin
o-criscione@acadnx:~/bin$ ./number_demo.bash
Enter a number > 2
Number is 2
Number is even
o-criscione@acadnx:~/bin$ ./number_demo.bash
Enter a number > 7
Number is 7
Number is odd
o-criscione@acadnx:~/bin$
```

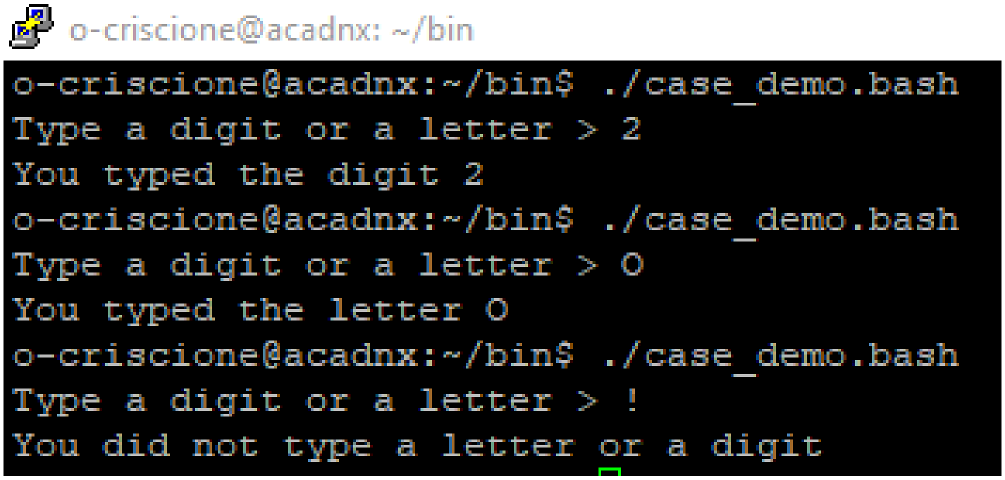
Exercise 11 Flow Control Part 2:

- Part A SS1:



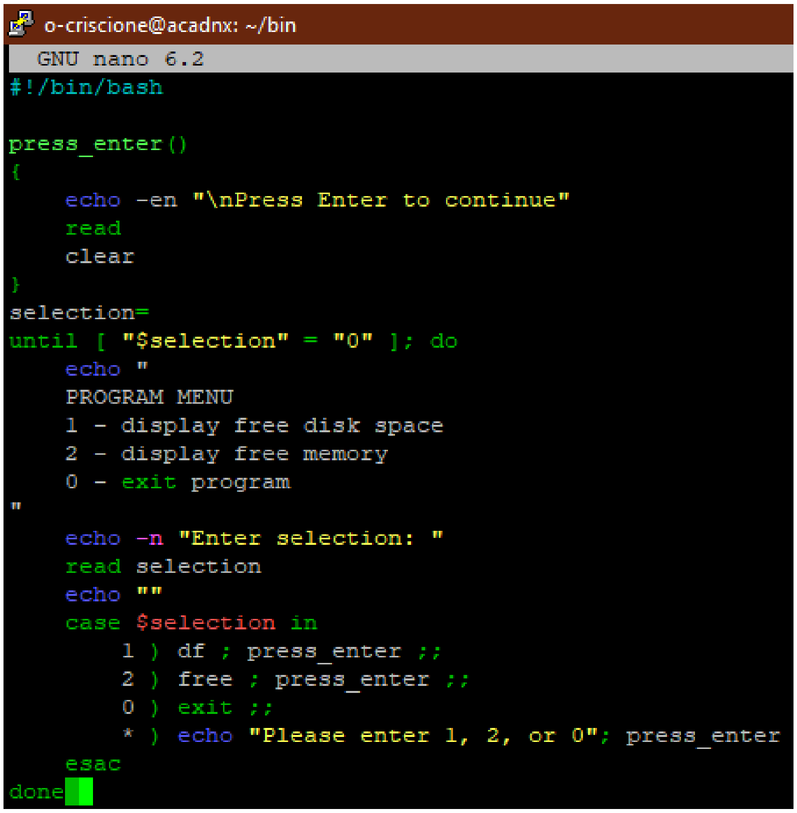
```
o-criscione@acadnx: ~/bin
  GNU nano 6.2
#!/bin/bash

echo -n "Type a digit or a letter > "
read character
case $character in
                                # Check for letters
    [[:lower:]] | [[:upper:]] ) echo "You typed the letter $character"
                                ;;
                                # Check for digits
    [0-9] )                     echo "You typed the digit $character"
;;
                                # Check for anything else
    * )                         echo "You did not type a letter or a digit"
esac
```

- Part A SS2:



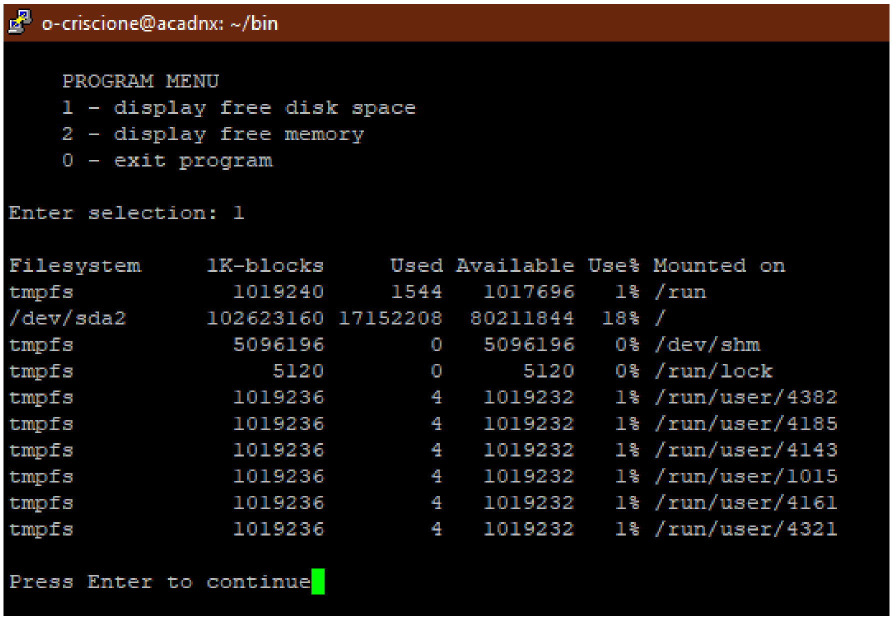
```
o-criscione@acadnx:~/bin$ ./case_demo.bash
Type a digit or a letter > 2
You typed the digit 2
o-criscione@acadnx:~/bin$ ./case_demo.bash
Type a digit or a letter > O
You typed the letter O
o-criscione@acadnx:~/bin$ ./case_demo.bash
Type a digit or a letter > !
You did not type a letter or a digit
```

- Part B SS1:



```
#!/bin/bash

press_enter()
{
    echo -en "\nPress Enter to continue"
    read
    clear
}
selection=
until [ "$selection" = "0" ]; do
    echo "
    PROGRAM MENU
    1 - display free disk space
    2 - display free memory
    0 - exit program
"
    echo -n "Enter selection: "
    read selection
    echo ""
    case $selection in
        1 ) df ; press_enter ;;
        2 ) free ; press_enter ;;
        0 ) exit ;;
        * ) echo "Please enter 1, 2, or 0"; press_enter
    esac
done
```

- Part B SS2:



```
    PROGRAM MENU
    1 - display free disk space
    2 - display free memory
    0 - exit program

Enter selection: 1

Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs            1019240     1544   1017696   1% /run
/dev/sda2      102623160 17152208  80211844  18% /
tmpfs            5096196        0   5096196   0% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs            1019236        4   1019232   1% /run/user/4382
tmpfs            1019236        4   1019232   1% /run/user/4185
tmpfs            1019236        4   1019232   1% /run/user/4143
tmpfs            1019236        4   1019232   1% /run/user/1015
tmpfs            1019236        4   1019232   1% /run/user/4161
tmpfs            1019236        4   1019232   1% /run/user/4321

Press Enter to continue
```

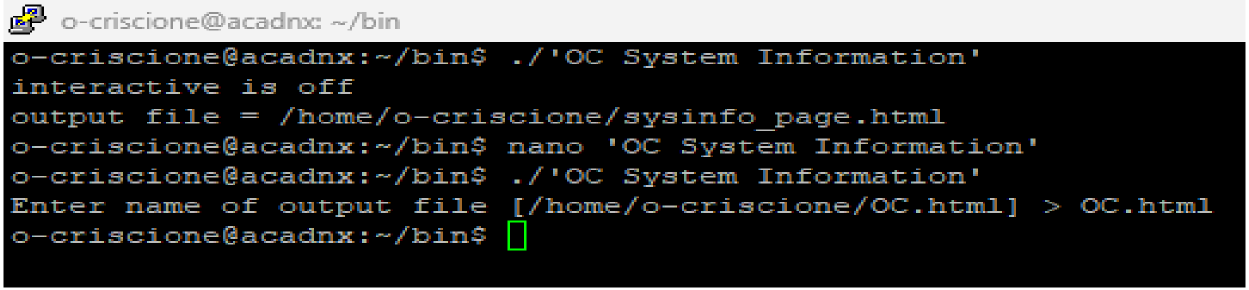
- Part C:

In the “loop_demo.bash” script, what would happen if we tested for the wildcard asterisk (*) first (i.e. we test in order *, 1, 2, 0)

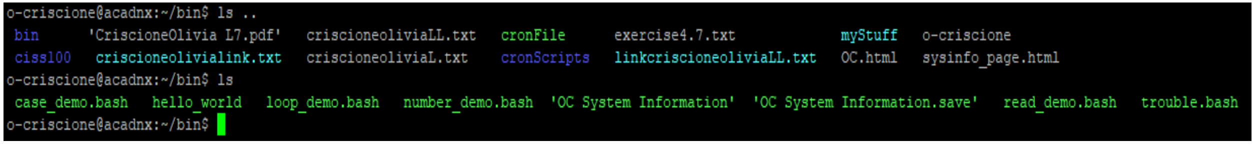
- Ans: If you enter an *, the program wouldn’t return a result because an * is not associated within the output options.

Exercise 12 Positional Parameters

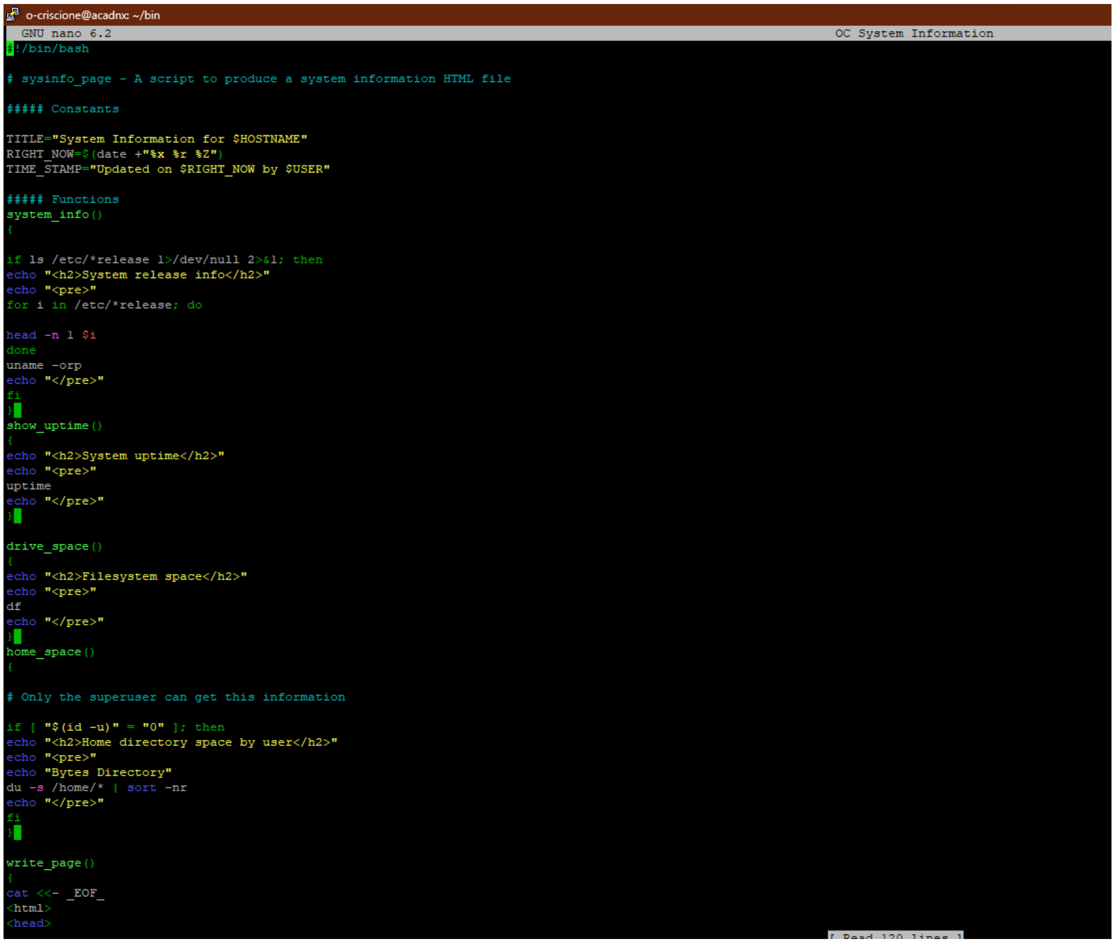
- Part A:



```
o-criscione@acadnx:~/bin$ ./'OC System Information'
interactive is off
output file = /home/o-criscione/sysinfo_page.html
o-criscione@acadnx:~/bin$ nano 'OC System Information'
o-criscione@acadnx:~/bin$ ./'OC System Information'
Enter name of output file [/home/o-criscione/OC.html] > OC.html
o-criscione@acadnx:~/bin$ 
```

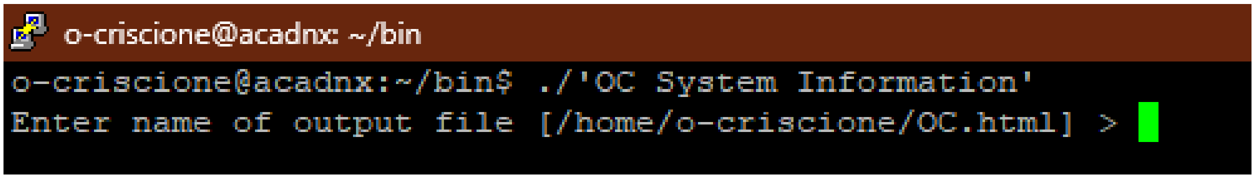


```
o-criscione@acadnx:~/bin$ ls ..
 bin      'CriscioneOlivia L7.pdf'   criscioneoliviaLL.txt   cronFile      exercise4.7.txt            myStuff   o-criscione
 ciss100   criscioneolivialink.txt   criscioneoliviaL.txt    cronScripts   linkcriscioneoliviaLL.txt  OC.html   sysinfo_page.html
o-criscione@acadnx:~/bin$ ls
 case_demo.bash   hello_world   loop_demo.bash   number_demo.bash  'OC System Information'  'OC System Information.save'   read_demo.bash   trouble.bash
o-criscione@acadnx:~/bin$ 
```

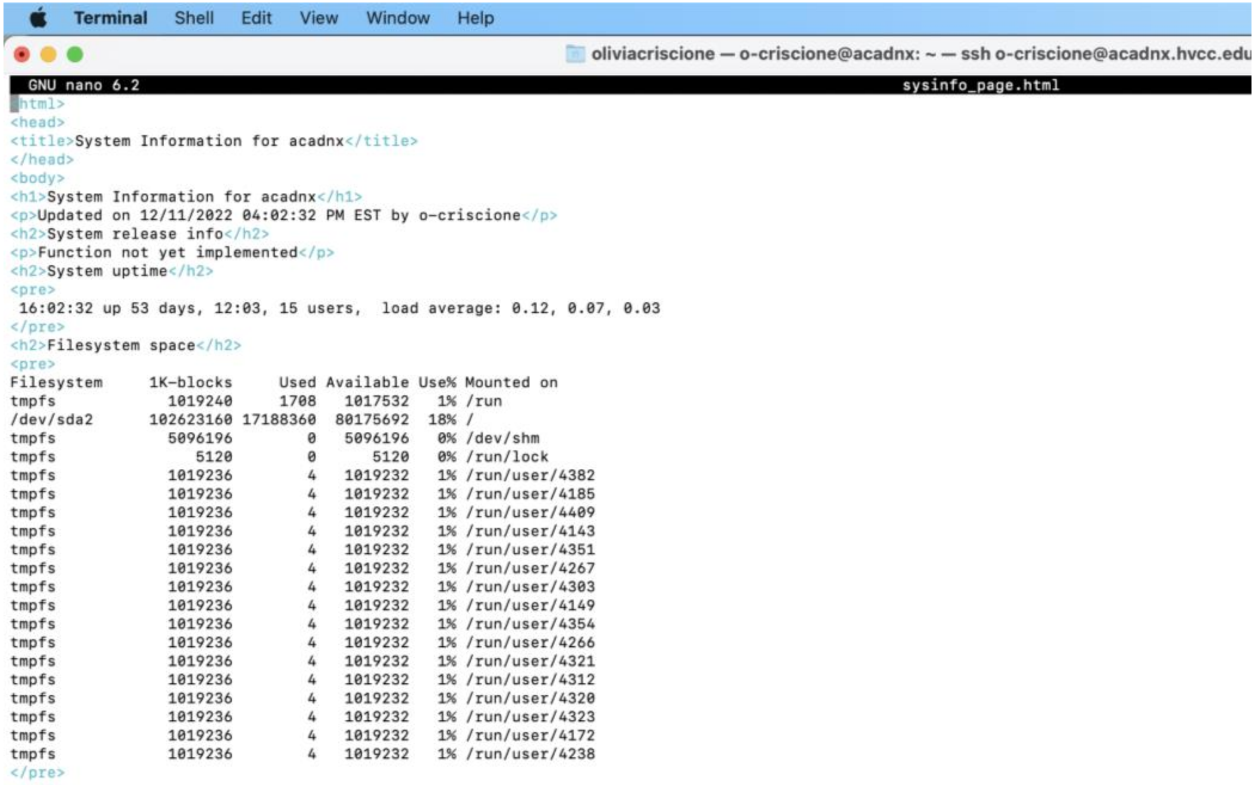


```
  GNU nano 6.2                                                                       OC System Information
#!/bin/bash

# sysinfo_page - A script to produce a system information HTML file

##### Constants

TITLE="System Information for $HOSTNAME"
RIGHT_NOW=$(date +"%x %r %Z")
TIME_STAMP="Updated on $RIGHT_NOW by $USER"

##### Functions
system_info()
{

if ls /etc/*release 1>/dev/null 2>&1; then
echo "<h2>System release info</h2>"
echo "<pre>"
for i in /etc/*release; do

head -n 1 $i
done
uname -orp
echo "</pre>"
fi
}
show_uptime()
{
echo "<h2>System uptime</h2>"
echo "<pre>"
uptime
echo "</pre>"
}

drive_space()
{
echo "<h2>Filesystem space</h2>"
echo "<pre>"
df
echo "</pre>"
}
home_space()
{

# Only the superuser can get this information

if [ "$(id -u)" = "0" ]; then
echo "<h2>Home directory space by user</h2>"
echo "<pre>"
echo "Bytes Directory"
du -s /home/* | sort -nr
echo "</pre>"
fi
}

write_page()
{
cat <<- _EOF_
<html>
<head>
                                                                     [ Read 120 lines ]
```

Exercise 13 Flow Control:



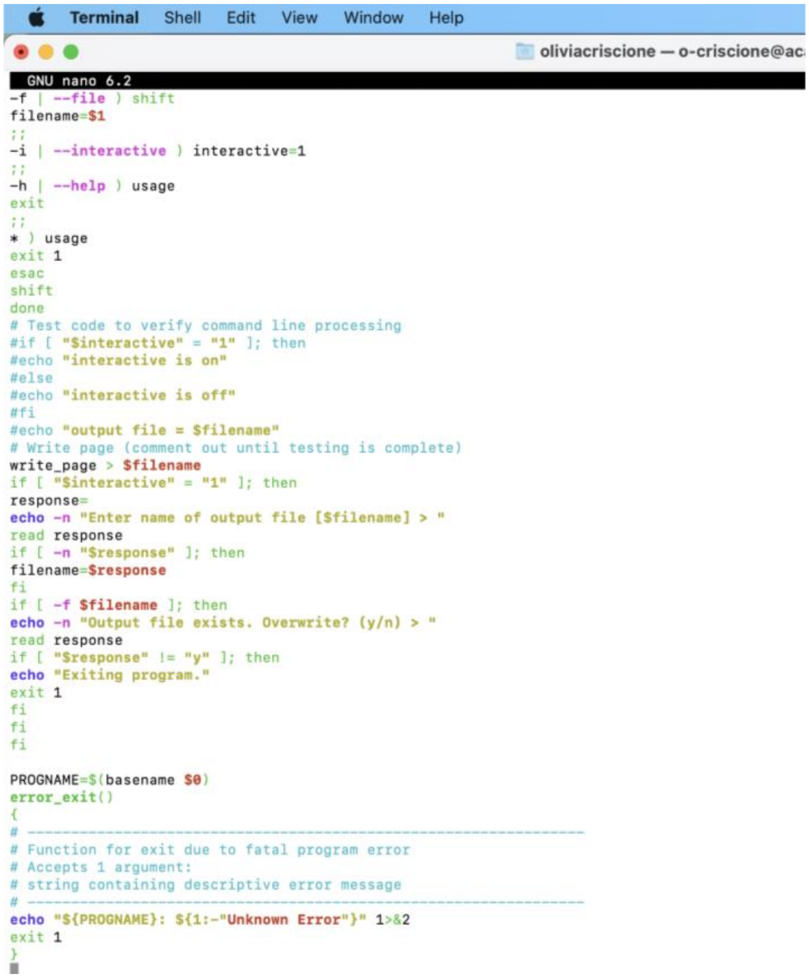
```
o-criscione@acadnx:~/bin$ ./'OC System Information'
Enter name of output file [/home/o-criscione/OC.html] >
```


Terminal
Shell
Edit
View
Window
Help
oliviacriscione — o-criscione@acadnx: ~ — ssh o-criscione@acadnx.hvcc.edu
GNU nano 6.2
sysinfo_page.html

```
<html>
<head>
<title>System Information for acadnx</title>
</head>
<body>
<h1>System Information for acadnx</h1>
<p>Updated on 12/11/2022 04:02:32 PM EST by o-criscione</p>
<h2>System release info</h2>
<p>Function not yet implemented</p>
<h2>System uptime</h2>
<pre>
 16:02:32 up 53 days, 12:03, 15 users,  load average: 0.12, 0.07, 0.03
</pre>
<h2>Filesystem space</h2>
<pre>
Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs            1019240     1708   1017532   1% /run
/dev/sda2      102623160 17188360  80175692  18% /
tmpfs            5096196        0   5096196   0% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs            1019236        4   1019232   1% /run/user/4382
tmpfs            1019236        4   1019232   1% /run/user/4185
tmpfs            1019236        4   1019232   1% /run/user/4409
tmpfs            1019236        4   1019232   1% /run/user/4143
tmpfs            1019236        4   1019232   1% /run/user/4351
tmpfs            1019236        4   1019232   1% /run/user/4267
tmpfs            1019236        4   1019232   1% /run/user/4303
tmpfs            1019236        4   1019232   1% /run/user/4149
tmpfs            1019236        4   1019232   1% /run/user/4354
tmpfs            1019236        4   1019232   1% /run/user/4266
tmpfs            1019236        4   1019232   1% /run/user/4321
tmpfs            1019236        4   1019232   1% /run/user/4312
tmpfs            1019236        4   1019232   1% /run/user/4320
tmpfs            1019236        4   1019232   1% /run/user/4323
tmpfs            1019236        4   1019232   1% /run/user/4172
tmpfs            1019236        4   1019232   1% /run/user/4238
</pre>
```

Exercise 14 Errors and Signals and Traps Part 1:

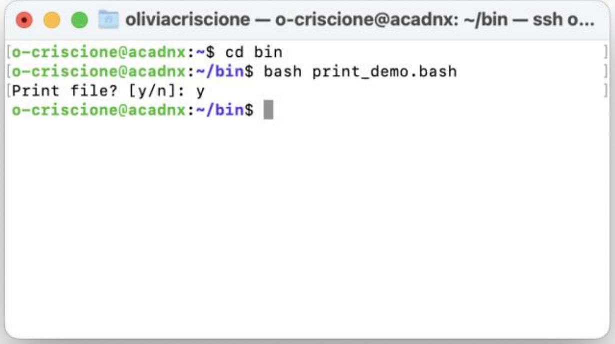
- Part A:



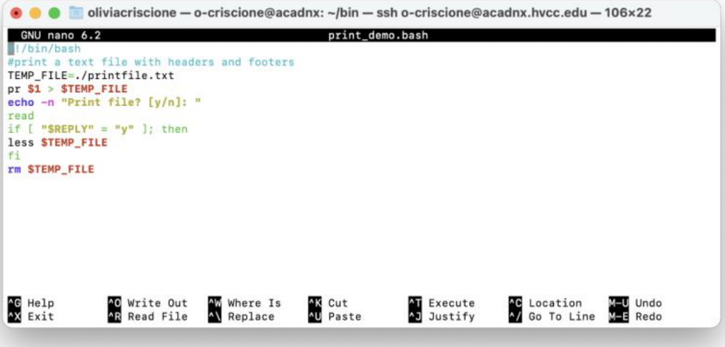
```
-f | --file ) shift
filename=$1
;;
-i | --interactive ) interactive=1
;;
-h | --help ) usage
exit
;;
* ) usage
exit 1
esac
shift
done
# Test code to verify command line processing
#if [ "$interactive" = "1" ]; then
#echo "interactive is on"
#else
#echo "interactive is off"
#fi
#echo "output file = $filename"
# Write page (comment out until testing is complete)
write_page > $filename
if [ "$interactive" = "1" ]; then
response=
echo -n "Enter name of output file [$filename] > "
read response
if [ -n "$response" ]; then
filename=$response
fi
if [ -f $filename ]; then
echo -n "Output file exists. Overwrite? (y/n) > "
read response
if [ "$response" != "y" ]; then
echo "Exiting program."
exit 1
fi
fi
fi

PROGNAME=$(basename $0)
error_exit()
{
# ----------------------------------------------------------------
# Function for exit due to fatal program error
# Accepts 1 argument:
# string containing descriptive error message
# ----------------------------------------------------------------
echo "${PROGNAME}: ${1:-"Unknown Error"}" 1>&2
exit 1
}
```

Exercise 15 Errors and Signals and Traps Part 2:

- Part A:



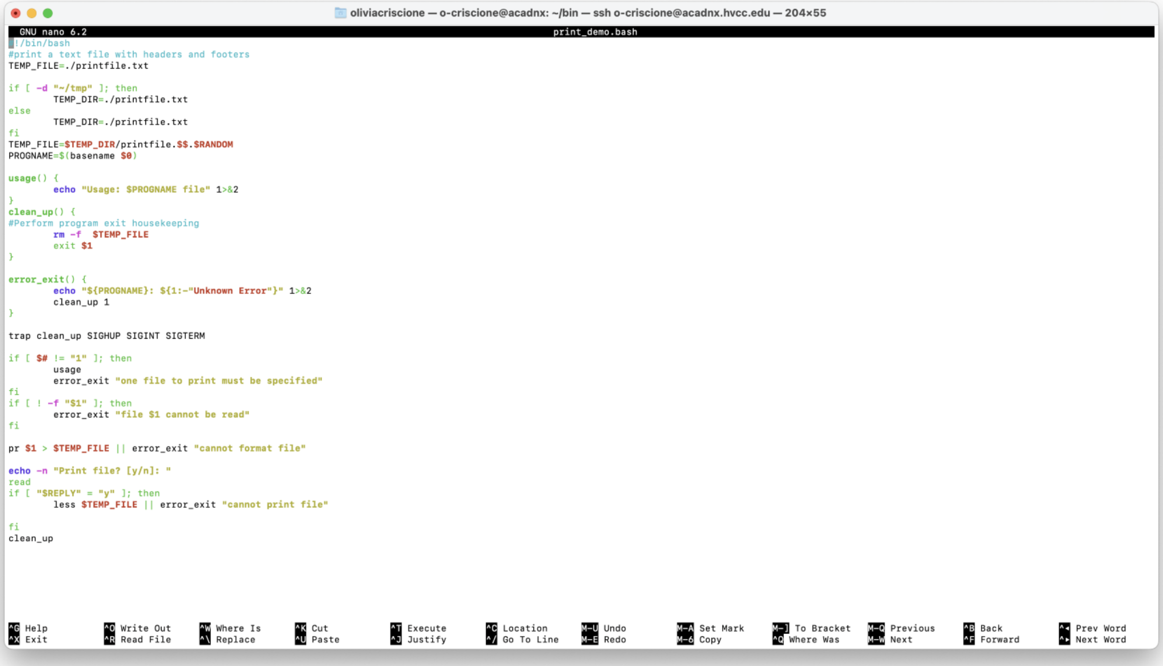
```
o-criscione@acadnx:~$ cd bin
o-criscione@acadnx:~/bin$ bash print_demo.bash
Print file? [y/n]: y
o-criscione@acadnx:~/bin$ 
```



```
#!/bin/bash
#print a text file with headers and footers
TEMP_FILE=./printfile.txt
pr $1 > $TEMP_FILE
echo -n "Print file? [y/n]: "
read
if [ "$REPLY" = "y" ]; then
less $TEMP_FILE
fi
rm $TEMP_FILE
```

Exercise 15 continued:

- Part D:



```
#!/bin/bash
#print a text file with headers and footers
TEMP_FILE=./printfile.txt

if [ -d "~/tmp" ]; then
        TEMP_DIR=./printfile.txt
else
        TEMP_DIR=./printfile.txt
fi
TEMP_FILE=$TEMP_DIR/printfile.$$.$RANDOM
PROGNAME=$(basename $0)

usage() {
        echo "Usage: $PROGNAME file" 1>&2
}
clean_up() {
#Perform program exit housekeeping
        rm -f  $TEMP_FILE
        exit $1
}

error_exit() {
        echo "${PROGNAME}: ${1:-"Unknown Error"}" 1>&2
        clean_up 1
}

trap clean_up SIGHUP SIGINT SIGTERM

if [ $# != "1" ]; then
        usage
        error_exit "one file to print must be specified"
fi
if [ ! -f "$1" ]; then
        error_exit "file $1 cannot be read"
fi

pr $1 > $TEMP_FILE || error_exit "cannot format file"

echo -n "Print file? [y/n]: "
read
if [ "$REPLY" = "y" ]; then
        less $TEMP_FILE || error_exit "cannot print file"

fi
clean_up
```