1. Finish the lab script.

Add another elif statement to check for symbolic link. If file is a symbolic link, then show me the original file that it points to.

Example output example:

calfile-link is a symbolic link.

The original file for calfile-link is : calfile

The size of calfile is : 453 bytes.

The calfile is owned by: <owner name>

Make sure it runs without errors.

```
 #! /bin/bash

# check directory
echo "Please enter a directory: "
read filetype
if [ -d $filetype ]
then
echo "$filetype is a directory"
echo "Do you want to see permissions of $filetype?"
read answer
  if [ $answer = "yes" ]
            then
              echo "doing listing of directory $filetype"
              ls -ld /etc
  fi
elif [ -f $filetype ]
then
echo "$filetype is a file, we will now check if it's executable"
echo -n "Do you want to see the permissions of: $filetype? "
read answer
if [ $answer = "yes" ]
  then
    echo "Doing listing of $filetype"
    ls -lh $filetype
fi
if [[ -L $filetype ]]
then
echo "$filetype is a symbolic link with `readlink -f $filetype` as the parent file"
parent=`readlink -f $filetype`
echo "$parent is `ls -l $parent |awk '{print $5}'` bytes large and belongs to `ls -l $parent | awk '{print $3}'`"
fi
fi
```

---

2. Comment every line in the script below and tell what each line is doing. Look up command in the man pages if you don't understand it. The script name is 'myargs'. Copy the script and run it on your system and give a detail comment of what each line is doing.

#!/bin/bash - **declares what shell runs the script**

Script name: myargs - **Script and file name.**

- **Extra line in comments**

Date: - **Date**

Name: - **Name**

Script to test command line arguments - **What the script does**

if [$# == 0] ; then - **If the value of # is equivalent to 0 then...**

  echo "Usage: $(basename $0) arg1 arg2 ... argn" \ - **Prints String with $(basename$0) being the filename without any of its leading directories**

  1>&2 **- Standard output goes to standard error**

  exit 1 - **Returns any syntax errors and exits**

fi - **End if statement**

echo "The name of this script is $0 ." - **Prints the name of the script like ./scripthere**

echo "The arguments are $* " - **Prints all the arguments**

echo "The first argument is $1" - **Prints the first argument**

echo "The second argument is $2" - **Prints the second argument**

echo "The number of arguments is $#" - **Prints the number of arguments**

previous_args=$* - **makes variable previous_args equal to all the arguments**

set niel khail nobo - **Sets 3 arguments as positional parameters**

echo "All the positional parameters are $*" - **Prints string and current arguments**

echo "The number of positional parameters is $#" - **Prints number of current arguments.**

echo $previous_args - **Prints the previous arguments through variable previous_args**

set -- **Unsets all positional parameters and since all arguments are gone, it instead will print all shell variables when called**

echo "Current args after -- are : $*" - **Prints current arguments**

set $previous_args - **Sets previous_args as the new positional parameters**

echo $* - **Prints all arguments of previous_args**

---

3. Copy the script and run it on your system and give a detail comment of what each line is doing.

#!/bin/bash - **Tells what shell runs the script**

Scriptname: tellage - **Comment on scriptname**

- **Extra line of comments**

read -p"How old are you? " - **prints String without line break and reads input**

age=$REPLY - **Set age to user's reply**

if ((age < 0 || age > 120)) - **If statement checking if age is less than 0 or more than 120**

then

    echo "You are not a real person! " - **Print String**

    exit 1 - **Return any syntax errors**

fi - **end if statement**

if ((age >= 0 && age < 13)) - **Another if statement following the clearance of the first**

then

    echo "You still have some of the best years of your life ahead." - **Print String**

elif ((age > 12 && age < 20)) - **Else if age does not match the parameters above, instead check these age groups..**

then

    echo "Important years to learn" - **Print String**

elif ((age >= 20 && age < 30)) - **Else if age does not match the parameters above, instead check these age groups..**

then

    echo "Time to find a potential mate!!" - **Print String**

elif ((age >= 30 && age < 40)) - **Else if age does not match the parameters above, instead check these age groups..**

then

    echo "You are probably changing diapers"

else - **If code passes without returning true from all the parameters above print the following**

    echo "Sorry I asked"

fi - **End if statement**