#!/usr/bin/sh

AGE="$0 -f directory

$0 -d directory

$0 -d -f directory

-f rename files

-d rename directories

"

usage ()

 {

 print -u2 "$USAGE"

 exit 1

 }

pathname ()

 {

 # function provided for the student

 print -- "${1%/\*}"

 }

basename ()

 {

 # function provided for the student

 print -- "${1##\*/}"

 }

find\_dirs ()

 {

 # function provided for the student

 find "$1" -depth -type d -name '\* \*' -print

 }

find\_files ()

 {

 # function provided for the student

 find "$1" -depth -type f -name '\* \*' -print

 }

my\_rename()

 {

 # the student must implement this function to my\_rename

 # $1 to $2

 # The following error checking must happen:

 # 1. check if the directory where $1 resided is writeable,

 # if not then report an error

 # 2. check if "$2" exists -if it does report and error and don't

 # do the mv command

 # 3. check the status of the mv command and report any errors

 #:

 # print "my\_rename: [$1] ==>> [$2]"

 echo "Trying: myrename $1 $2"

 dir=`dirname "$1"`

 if [ ! -w "$dir" ]; then

 echo "$dir: Directory not writable"

 return

 fi

 if [ -e "$2" ]; then

 echo "$2: Exists, cannot rename"

 else

 mv "$1" "$2"

 if [ $? != 0 ]; then

 echo "mv: failed, return code=$?"

 fi

 fi

 }

fix\_dirs ()

 {

 # The student must implement this function

 # to actually call the my\_rename funtion to

 # C

 # changing all of the spaces to -'s

 # if the name were "a b", the new name would be a-b

 # if the name were "a b" the new name would be a----b

 #:

 # changing all of the spaces to -'s

 # if the name were "a b", the new name would be a-b

 # if the name were "a b" the new name would be a----b

 #:

 echo "Fixing dirs starting at: $1 ..."

 #find\_dirs "$1"

 IFS=";"

 set -A dirs "$(find\_dirs "$1")"

 for dirpath in "${dirs[@]}"

 do

 parent=`dirname "$dirpath"`

 olddirname=`basename "$dirpath"`

 newdirname=`print $olddirname | tr ' ' '-'`

 olddir=$dirpath

 newdir="$parent/$newdirname"

 #print "[$olddir] ==>> [$newdir]"

 my\_rename "$olddir" "$newdir"

 done

 }

fix\_files ()

 {

 # The student must implement this function

 # to actually call the my\_rename funtion to

 # change the name of the file from having spaces to

 # changing all of the spaces to -'s

 # if the name were "a b", the new name would be a-b

 # if the name were "a b" the new name would be a----b

 #:

 echo "Fixing files under $1 ..."

 IFS=";"

 set -A files "$(find\_files "$1")"

 # echo ${files[0]}

 # echo ${files[1]}

 # echo ${files[2]}

 for fileparts in "${files[@]}"

 do

 IFS="|"

 set -A parts $fileparts

 #print "[${parts[0]} -- ${parts[1]}]"

 dir=${parts[0]}

 oldfilename=${parts[1]}

 newfilename=`print $oldfilename | tr ' ' '-'`

 #print "[$newfilename]"

 oldfile="$dir/$oldfilename"

 newfile="$dir/$newfilename"

 #print "[$oldfile] ==>> [$newfile]"

 my\_rename "$oldfile" "$newfile"

 done

 }

WFILE=

WDIR=

DIR=

if [ "$#" -eq 0 ]

 then

 usage

 fi

while [ $# -gt 0 ]

 do

 case $1 in

 -d)

 WDIR=1

 ;;

 -f)

 WFILE=1

 ;;

 -\*)

 usage

 ;;

 \*)

 if [ -d "$1" ]

 then

 DIR="$1"

 print

 else

 echo "$1 does not exist ..."

 exit 1

 fi

 ;;

 esac

 shift

 done

# The student must implement the following:

# - if the directory was not specified, the script should

# print a message and exit

if [ "$DIR" = "" ]

then

 echo "Directory was not specified"

 exit 1

fi

# - if the Directory specified is the current directory, the script

# print a error message and exit

if [ "$DIR" = "$PWD" ]

then

 echo "Directory '$DIR' (current directory) is not allowed"

 exit 1

fi

# - if the directory specified is . or .. the script should print

# an error message and exit

if [[ "$DIR" = "." || "$DIR" = ".." ]]

then

 echo "Directory '.' or '..' is not allowed"

 exit 1

fi

# - if both -f and -d are not specified, the script should print a

# message and exit

#

if [[ "$WDIR" = "" && "$WFILE" = "" ]]

then

 echo "Neither -f nor -d was specified; not allowed"

 exit 1

fi

if [ "$WDIR" -a "$WFILE" ]

 then

 fix\_files "$DIR"

 fix\_dirs "$DIR"

elif [ "$WDIR" ]

 then

 fix\_dirs "$DIR"

elif [ "$WFILE" ]

 then

 fix\_files "$DIR"

 fi