The students are required to implement the areas in bold

Please note that his script was written for the KornShell - if you implement it with bash, you may need to change the print commands to echo commands depending on your bash implementation.

```
#!/usr/bin/ksh

USAGE="$0 -f directory
$0 -d  directory
$0 -d -f directory

-f my_rename files
-d my_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
    :   # remove this line when you add your code
    }

fix_dirs ()
    {
    # The student must implement this function
    # to actually call the my_rename funtion to
    # change the name of the directory 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
    :   # remove this line when you add your code

    }

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
    :   # remove this line when you add your code

    }

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"
      else
          print -u2 "$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 the Directory specified is the current directory, the script
#   print a error message and exit

# - if the directory specified is . or .. the script should print
#   an error message and exit

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

if [ "$WDIR" -a "$WFILE" ]
    then
    fix_files "$DIR"
    fix_dirs "$DIR"
elif [ "$WDIR" ]
    then
    fix_dirs "$DIR"
elif [ "$WFILE" ]
    then
    fix_files "$DIR"
    fi
```