

Practical Exercises 7 : Reading and Formatting

```
$ cat mycp
#
# Copy a file
#
numargs=$#
# save this for later use
filelist=
copylist=
#
# Process the arguments, storing all but the last in filelist
#
while [ "$#" -gt 1 ]
do
    filelist="$filelist $1"
    shift
done
to="$1"
#
# If less than two args, or if more than two args and last arg
# is not a directory, then issue an error message
#
if [ "$numargs" -lt 2 -o "$numargs" -gt 2 -a ! -d "$to" ]if [ -d "$to" ]
then
    echo "Usage: mycp file1 file2"
    echo "      mycp file(s) dir"
    exit 1
fi
#
# Sequence through each file in filelist
#
for from in $filelist
do
    #
    # See if destination file is a directory
    #
    if [ -d "$to" ]
    then
        tofile="$to/${basename $from}"
    else
        tofile="$to"
    fi
    #
    # Add file to copylist if file doesn't already exist
    # or if user says it's okay to overwrite
    #
    if [ -e "$tofile" ]
    then
        echo "$tofile already exists; overwrite (yes/no)? \c"
```

```

        read answer
        if [ "$answer" = yes ]
        then
            copylist="$copylist $from"
        fi
    else
        copylist="$copylist $from"
    fi
done
## Now do the copy -- first make sure there's something to copy
#
if [ -n "$copylist" ]
then
    cp $copylist $to
    # proceed with the copy
fi

```

```

-----
$ ls          # See what's around
$ ls bin      # And what's in bin?
$ mycp        # No arguments
Usage: mycp file1 file2
mycp file(s) dir
$ mycp names prog1 prog2
Last arg isn't a directory
Usage: mycp file1 file2
mycp file(s) dir
$ mycp names prog1 prog2 lu bin
Legitimate use
bin/prog1 already exists; overwrite (yes/no)? yes
bin/lu already exists; overwrite (yes/no)? no
$ ls -l bin   # See what happened

```

```

$ cat rolo
#
# rolo - rolodex program to look up, add, and
#
#
remove people from the phone book#
# Display menu
#
echo '
    Would you like to:

    1. Look someone up
    2. Add someone to the phone book
    3. Remove someone from the phone book

    Please select one of the above (1-3): \c'
#

```

```

# Read and process selection
#
read choice
echo
case "$choice"
    in
        1) echo "Enter name to look up: \c"
            read name
            lu "$name";;  ### programs lu, rem, and add are from earlier exercises
        2) echo "Enter name to be added: \c"
            read name
            echo "Enter number: \c"
            read numberadd "$name" "$number";;
        3) echo "Enter name to be removed: \c"
            read name
            rem "$name";;
        *) echo "Bad choice";;
esac

```

```

$ cat addi
#
# add pairs of integers on standard input
#
while read n1 n2
do
    echo $((n1 + n2))
done

```

```

-----
$ addi
10 25
35
-5 12
7
123 3
126
Ctrl+d

```

```

$ cat data
1234 7960
593 -595
395 304
3234 999
-394 -493
$ addi < data > sums
$ cat sums
9194
-2
699
4233

```

```
cat align
#
# Align two columns of numbers
# (works for numbers up to 12 digits long, including sign)
cat $* |
while read number1 number2
do
    printf "%12d %12d\n" $number1 $number2
done
```

```
-----
$ cat  data
1234 7960
 593 -595
 395  304
3234  999
-394 -493
```

```
$ align data
      1234      7960
      593      -595
      395       304
     3234       999
     -394      -493
```
