```
#! /bin/sh
1
2
3
  #declarations
4 ZIP_ARCHIVES="Complaints/Zip_Archives"
5 FILE ARCHIVES="Complaints/File Archives"
  PROCESSING="Complaints/Processing"
  MAIN DIRECTORY="Complaints"
  MASTER FILE="Complaints/Complaints Master.csv"
8
9
  #ensure in home directory
10
  cd ∼
11
12
  setup(){
13
14
15
       #make or remake the main folder structure
       if [[ -d $MAIN DIRECTORY ]];then
16
           rm -r $MAIN DIRECTORY
17
           mkdir -p "$FILE ARCHIVES"
18
           mkdir -p "$ZIP_ARCHIVES"
19
           mkdir -p "$PROCESSING"
20
       else
21
           mkdir -p "$FILE_ARCHIVES"
22
           mkdir -p "$ZIP ARCHIVES"
23
           mkdir -p "$PROCESSING"
24
       fi
25
26
       #setup master file with header
27
       if ! [[ -f $MASTER FILE ]];then
28
           echo "Complaint ID,Date Received,Company,Product,Issue" >
29
   $MASTER_FILE
...
30
       fi
31
32
  }
33
34
  option 1(){
35
36
       #tar -xzf aggregate_complaints_001.tar.gz -C $FILE_ARCHIVES #THIS IS FOR
37
   TESTING ONLY
38
       #create a list of all zip files to process
39
       active_zip_file_list=("aggregate_complaints"*)
40
41
       #unpack each zip file then move the zip file to the zip archive
42
       for i in ${active zip file list[*]};
43
           do
44
               tar -xzf $i -C "$PROCESSING"
45
               cp $i "$ZIP ARCHIVES/$i"
46
           done
47
```

```
48
       #notify the user of a wait
49
       echo "Please wait, processing "${#active_zip_file_list[*]}" zipped
50
   complaint file(s)"
...
51
       #populate arrays of file names in processing and in file archives
52
  WARNING: THESE ARE FULL PATHS
       active_file_list=("$PROCESSING"/*)
53
       archive file list=("$FILE ARCHIVES"/*)
54
55
       #ensure that we don't process a file that has previously been processed
56
       for i in ${active_file_list[*]};
57
           do
58
               for j in ${archive_file list[*]};
59
                    do
60
                        i=(\$(echo \$i \mid cut -d "/" -f 3)) #extracting just the
61
   file name
                        j=(\$(echo \$j \mid cut -d "/" -f 3))
                                                              #extracting just the
62
   file name
                        if [[ $i == $j ]];then
63
                        rm $PROCESSING/$i
                                                              #keep in mind that
64
   $i is now just the file name
                        echo $i" deleted"
65
                        fi
66
                    done
67
           done
68
69
70
       #process all files in processing, move them to file archives as they are
71
   processed
...
       #issues: skipping first line, last line maybe
72
       active file list=("$PROCESSING"/*)
73
       for i in ${active file list[*]};
74
75
       do
           \#i=(\$(echo \$i \mid cut -d "/" -f 3)) \#extracting just the file name
76
           #echo $i
77
78
           while IFS= read -r line
79
           do
80
               linelen=${#line}
81
               if [[ linelen -eq 2 ]]; then #this should work to catch the
82
   last line as well
                    continue
83
               fi
84
               line=$(echo $line | tr " " " ")
85
               mya=($(echo $line | cut -d ":" -f 1- --output-delimiter=" "))
86
87
               #get complaint id
88
               cid=${mya[-1]}
89
```

```
cid=$(echo $cid | cut -d "\"" -f2)
90
91
                 #get date received
92
                 date_received=${mya[1]}
93
                 date_received=$(echo $date_received | cut -d "\"" -f2)
94
95
96
                 #get company WARNING company is messed up
                 company=\{mya[-11]\}
97
                 company=$(echo $company | tr "," " ")
98
                 company=$(echo $company | cut -d "\"" -f2)
99
100
                 #get product
101
                 product=${mya[2]}
102
                 product=$(echo $product | tr "," " ")
103
                 product=$(echo $product | cut -d "\"" -f2)
104
105
106
                 #get issue
107
                 issue=\$\{mya[4]\}
108
                issue=$(echo $issue | tr "," " ")
109
                 issue=$(echo $issue | cut -d "\"" -f2)
110
111
                 #write the data to the file
112
                 output="$cid,$date received,$company,$product,$issue"
113
                 echo $output >> $MASTER FILE
114
115
                #read < /dev/tty TESTING PURPOSES ONLY</pre>
116
            done < $i
117
118
            #move the processed file
119
            i=(\$(echo \$i \mid cut -d "/" -f 3))
120
            mv "$PROCESSING/$i" "$FILE ARCHIVES/$i"
121
122
        done
123
124
   }
125
126
   option_2(){
127
        #cleanup
128
129
        #load an array of all file records
130
        while IFS= read -r line
131
            do
132
                #echo $line
133
                if [[ $line == "Complaint ID,Date
134
   Received, Company, Product, Issue" ]]; then #skip the header
 ...
                     continue
135
                 fi
136
137
```

```
cid_list+=($(echo $line | cut -f 1-))
138
139
                 #read < /dev/tty TESTING PURPOSES ONLY</pre>
140
            done < $MASTER FILE
141
142
            current count=${#cid list[*]}
143
            echo "Number of current records: $current count"
144
145
            cid_list=($(
146
                 for i in ${cid list[*]};
147
148
                     echo $i
149
                 done | sort -u))
150
151
            post count=${#cid list[*]}
152
            echo "Number of records after removing duplicates: $post count"
153
            echo "Duplicate records removed: $(( $current count - $post count
154
 ...
155
            #now write that new list to the csv file (but use a duplicate for
156
   testing)
            echo "Complaint ID,Date Received,Company,Product,Issue" >
157
    "$MASTER FILE"
 ...
            for i in ${cid list[*]};
158
159
                 echo $i >> "$MASTER_FILE"
160
            done
161
162
            read -p "Press Enter to continue: "
163
164
   }
165
166
   option 3(){
167
        #reporting
168
        clear
169
        #initialize list and load all records into csv loaded list
170
        while IFS= read -r line
171
            do
172
                #echo $line
173
                 if [[ $line == "Complaint ID,Date
174
   Received, Company, Product, Issue" ]]; then #skip the header
                     continue
175
                 fi
176
177
                 csv loaded+=($(echo $line | cut -f 1-))
178
179
                 #read < /dev/tty TESTING PURPOSES ONLY</pre>
180
            done < $MASTER FILE
181
182
```

```
#load products into product_list
183
        for i in ${csv loaded[*]};
184
185
            product=$(echo $i| cut -d "," -f4)
186
            product list+=($product)
187
188
        done
189
        #solicit products and show reporting
190
        while true:
191
        do
192
            matches=1
193
            issue_list=()
194
            #product_list=()
195
            company_list=()
196
            clear
197
            echo "AVAILABLE PRODUCTS"
198
            echo "----
199
200
            #remove duplicate products
201
            product_list=($(
202
                 for i in ${product_list(*)};
203
                 do
204
                     echo $i
205
                 done | sort -u))
206
207
            #list the products
208
            for i in ${!product_list[*]};
209
            do
210
                 i=\$((\$i+1))
211
                product=$(echo ${product_list[$i-1]} | tr "_" " ")
212
                 product=$(echo $product | sed 's/ / /g')
213
                 if [[ $i -lt 10 ]]; then
214
                     echo " $i $product"
215
                 else
216
                     echo "$i $product"
217
                 fi
218
            done
219
220
            #get the user product number choice
221
            echo
222
            read -p "Enter the product number (zero to exit): " choice
223
            my_count=${#product_list[*]}
224
            #error trapping for correct input
225
            #echo $choice
226
            if [[ $choice -eq 0 ]]; then
227
228
                 break
            fi
229
230
            #get the proper product from the product list, based on the user's
231
```

```
numerical input
231...
             choice text=${product list[$choice -1]}
232
233
             #get the matching issues and companies
234
             for i in ${csv loaded[*]};
235
236
                 product=$(echo $i | cut -d "," -f4)
237
                 if [[ $product == $choice text* ]];then
238
                     issue=$(echo $i | cut -d "," -f5)
239
                     issue list+=($issue)
240
                     company=$(echo $i | cut -d "," -f3)
241
                     company_list+=($company)
242
                 fi
243
             done
244
245
             #get a count of matching records (issue list length with dupes)
246
             matches=${#issue list[*]}
247
248
             #remove duplicates from the issue list
249
             issue_list=($(
250
             for i in ${issue list[*]};
251
             do
252
                 echo $i
253
             done | sort -u))
254
255
             #remove duplicates from the company list
256
             company list=($(
257
             for i in ${company list[*]};
258
             do
259
                 echo $i
260
             done | sort -u))
261
262
             #show the report elements
263
             clear
264
             product=$(echo $choice_text | tr [:lower:] [:upper:])
265
             choice text=$(echo $choice text | tr [:lower:] [:upper:])
266
             choice text=$(echo $choice text | tr " " " ")
267
             choice_text=$(echo $choice_text | sed 's/ / /g')
268
             echo "PRODUCT: $choice text"
269
             echo "Number of companies involved: ${#company list[*]}"
270
             echo " Number of matching records: $matches"
271
             echo
272
             echo "
                                           ISSUES"
273
             echo "
274
             for i in ${issue list(*)};
275
276
                 i=$(echo $i | tr "_" " ")
277
                 echo $i
278
279
             done
```

```
280
            echo
            read -p "Press enter to continue" dog
281
282
        done #end of while true
283
   }
284
285
286
287
288
   #setup
289
290
291
   while true;
292
   do
293
        clear
        echo "---- MAIN MENU -----
294
295
   Please select from the following options:
296
297
        Process Complaint Files
298
   1.
   2.
        Remove Duplicate Complaint Records
299
        Report by Product
   3.
300
   4.
        Exit
301
302
303
        read -p "Option#: " user menu choice
304
305
        if [[ user_menu_choice -eq 1 ]];then
306
            option 1
307
            continue
308
        elif [[ user_menu_choice -eq 2 ]];then
309
            option_2
310
            continue
311
        elif [[ user menu choice -eq 3 ]]; then
312
            option 3
313
            continue
314
        elif [[ user menu choice -eq 4 ]];then
315
            clear
316
            break
317
        else
318
            clear
319
            echo "That is not a valid option. Please press Enter to try again."
320
            read input
321
            clear
322
            continue
323
        fi
324
   done
325
326
```