



**Faculty of Engineering & Technology – Electrical & Computer
Engineering Department**

First Semester 2021 – 2022

Linux Laboratory – ENCS3130.

Shell Script Project (Project 1)

Name: Ahmaide Al-Awawdah.

Name: Khalid Addabe

ID: 1190823

ID: 1190507

Section: 1

Instructor: Dr. Mohammad Jubran

TA: Eng.Katy Sadi

Date: 7th – 29th November 2021

1. Abstract

This is project which the code is written in shell script to do operations on a contact file that contains user's data (first name, last name, phone numbers, and email) as shown in the figure below.



Figure 1: The Contact File

The user can do operations such as adding a new contact, deleting a contact, printing all contacts with chosen fields, searching for a contact, and editing a contact, from a menu that should be displayed on the screen as the figure below.

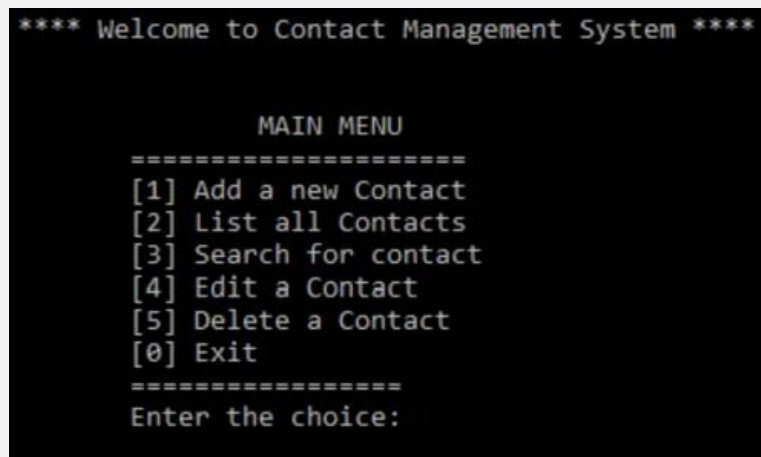


Figure 2: Operations Menu

❖ Table of Content

1. Abstract.....	2
2. The Code	4
2.1. The main menu.....	4
2.2. The First Function: Adding New Contact.....	4
2.3. The Second Function: Display All Contacts	6
2.4. The Third Function: Search for a Contact.....	7
2.5. The fourth Function: Edit a Contact	7
2.5.1.....	8
2.5.1. The First Name Edit Function.....	8
2.5.2. The Last Name Edit Function	8
2.5.3. The Phone Number Edit Function.....	9
2.5.4. The Email Change Function	10
2.6. The Fifth Function: Deleting a contact	10
3. Testing Examples.....	11
3.1. Test Example 1	11
3.2. Test Example 2	13
4. Conclusion.....	15
5. Appendix.....	16

2. The Code

The code of the project is divided into several functions, each function does a specific task.

The whole code can be found in the [Appendix](#).

2.1. The main menu

The part where the code starts compiling from where it calls each function only at it's case depending on the user's choice after displaying the menu, but first it needs to ask the user for the file name to see if it is available.

```
553 #===== Main menu that first appears on the user screen =====
554
555 read -p "Enter File Name: " f          # f is a variable of the file that will contain the data of users
556
557 if [ -e $f ]                          # Only do the following if the file exists
558 then
559     select=0                          # The user menu at first
560     printf "**** Welcome to Contact Management System ****\n\n"
561     printf "Main Menu\n"
562     printf "=====\n"
563     printf "[1] Add a new Contact\n"
564     printf "[2] List all Contacts\n"
565     printf "[3] Search for Contact\n"
566     printf "[4] Edit a Contact\n"
567     printf "[5] Delete a Contact\n"
568     printf "[0] Exit\n"
569     read -p "Enter the choice: " select # Here the user enters his selection
570     m=(wc -l "$f")
571     while [ $select -ge 1 -a $select -le 5 ] # a loop so the menu keeps appearing until the user asks it to stop
572     do
573         case "$select"
574         in
575             1) first ;;          # If select = 1, go the (first) function
576             2) second ;;         # If select = 2, go the (second) function
577             3) third ;;          # If select = 3, go the (third) function
578             4) fourth ;;         # If select = 4, go the (fourth) function
579             5) fifth ;;          # If select = 5, go the (fifth) function
580             *) break ;;         # In other cases, end the loop
581         esac
582         printf "**** Welcome to Contact Management System ****\n\n"
583         printf "Main Menu\n"
584         printf "=====\n"
585         printf "[1] Add a new Contact\n"
586         printf "[2] List all Contacts\n"
587         printf "[3] Search for Contact\n"
588         printf "[4] Edit a Contact\n"
589         printf "[5] Delete a Contact\n"
590         printf "[0] Exit\n"
591         read -p "Enter the choice: " select
592     done
593 else
594     echo "Error the file you entered doesn't exist" # for the file not found case
595 fi
```

2.2. The First Function: Adding New Contact

This Function is for when the user wants to enter a new contact, the user should at least enter a first name and one phone number, the program gives the user a choice to enter a last name or an email, and all the names should only consist of alphabetical letters, the phone numbers should only contain numbers from range 9-10 digits, and the email should include only one @ if the user enters any wrong data the program will ask for the data to be reentered.

```

177 ##### The function that will add a new contact #####
178 first () {
179     printf "\n"
180     #----- Adding First name -----
181     notChar="[A-Za-z]+$"
182     chooser=0
183     while [ $chooser -eq 0 ]
184     do
185         read -p "Enter first name: " fn          # The user will enter the new first name
186         if ! [[ $fn == $notChar ]]                # The name must only contain letters
187         then
188             echo "That's not a valid name, try again"
189         else
190             chooser=1
191         fi
192     done
193     allInfo=$fn
194     allInfo+=", "
195     #----- Adding a last name by the user's choice -----
196     read -p "Do you want to enter a last name (if yes enter 1, other if not): " chooser
197     if [ $chooser = "1" ]
198     then
199         notChar="[A-Za-z]+$"
200         chooser=0
201         while [ $chooser -eq 0 ]
202         do
203             read -p "Enter last name: " ln          # The user will enter the new last name
204             if ! [[ $ln == $notChar ]]                # The name must only contain letters
205             then
206                 echo "That's not a valid name, try again"
207             else
208                 chooser=1
209             fi
210         done
211         allInfo+=$ln
212         allInfo+=", "
213     fi
214     #----- Adding phone number/s -----
215     read -p "Enter the number of phone numbers: " nop
216     co=1
217     allp=""
218     while [ $co -le $nop ]
219     do
220         read -p "Enter phone $co : " pn
221         re="^[0-9]+$"
222         if ! [[ $pn == $re ]]
223         then
224             echo "The Number you entered isn't valid (contains other things), Please try again."
225         else
226             checking=$(echo -n $pn | wc -c)
227             if [ $checking -ge 9 -a $checking -le 10 ]
228             then
229                 alreadyExists=$(grep "$pn" $f)
230                 if [ "$alreadyExists" ]
231                 then
232                     echo "This number already exists"
233                     if [ $nop -eq 1 ]
234                     then
235                         echo "In this case the system won't let the user enter a new number instead unless it was the only number"
236                         echo "you need to enter a new phone number!"
237                     fi
238                     nop=$(( $nop - 1 ))
239                     # So the number of the entire numbers will decrease if the user entered an existing number
240                     # The case of the number is a new number
241                     allp+=$pn
242                     if [ $co -ne $nop ]
243                     then
244                         allp+=";"
245                     fi
246                     co=$(( $co + 1 ))
247                 else
248                     # The case of that the length of the number is not between 9-10
249                     echo "The Number you entered isn't valid (size issue), Please try again."
250                 fi
251             fi
252         done
253         allInfo+=$allp
254         allInfo+=", "
255     #----- Adding an email by the user's choice -----
256     read -p "Do you want to enter an email (if yes enter 1, other if not): " chooser
257     if [ $chooser = "1" ]
258     then
259         chooser=0
260         at="@"
261         while [ $chooser -eq 0 ]
262         do
263             read -p "Enter Email: " e
264             if [[ "$e" == *$at* ]]
265             then
266                 C=$(echo $e | sed 's/[@]//g')
267                 CO=$(echo $C | wc -c)
268                 if [ $CO -eq 1 ]
269                 then
270                     chooser=1
271                 else
272                     echo "The email you entered isn't valid!"
273                 fi
274             else
275                 echo "The email you entered isn't valid!"
276             fi
277         done
278         allInfo+=$e
279     fi
280     #----- Add all the info in a new line to the contact file -----
281     echo $allInfo >> $f
282     printf "\n"
283     printf "\n"
284     printf "\n"
285 }
286

```

2.3. The Second Function: Display All Contacts

This function is to display all contacts where the user chooses if the field will be displayed or not, if the field is empty the system will print “Empty” and the user can choose to not print any field.

```
288 #----- The function that will print the contact data -----
289 second () {
290     #----- Here the user selects what to print and the if statment works in binary bits -----
291     printf "\n"
292     caser=0
293     read -p " If you want the first name to be printed enter 1, enter others to not: " chooser
294     if [ $chooser = "1" ]
295     then
296         caser=$(( $caser + 8 ))
297     fi
298     read -p " If you want the last name to be printed enter 1, enter others to not: " chooser
299     if [ $chooser = "1" ]
300     then
301         caser=$(( $caser + 4 ))
302     fi
303     read -p " If you want the phone numbers to be printed enter 1, enter others to not: " chooser
304     if [ $chooser = "1" ]
305     then
306         caser=$(( $caser + 2 ))
307     fi
308     read -p " If you want the email to be printed enter 1, enter others to not: " chooser
309     if [ $chooser = "1" ]
310     then
311         caser=$(( $caser + 1 ))
312     fi
313     m=0
314     printf "\n"
315     counter=0
316     #----- A loop to go around all the lines -----
317     while read line;
318     do
319         printf "\n"
320         printf " "
321         if [ $caser -ge 8 ]
322         then
323             if [ $counter -eq 0 ]
324             then
325                 printf "First name"
326             else
327                 n1=$( echo $line | cut -d "," -f1 )
328                 printf $n1
329             fi
330             m=1
331             # m=1 means that there is a previous data printed in order to add a coma
332             mod=$(( $caser % 8 ))
333             if [ $mod -ge 4 ]
334             then
335                 if [ $m -eq 1 ]
336                 then
337                     printf ", "
338                 fi
339                 if [ $counter -eq 0 ]
340                 then
341                     printf "Last name"
342                 else
343                     n1=$( echo $line | cut -d "," -f2 )
344                     if [ $n1 ]
345                     then
346                         printf $n1
347                     else
348                         printf "Empty"
349                     fi
350                 fi
351                 m=1
352             fi
353             mod=$(( $caser % 4 ))
354             if [ $mod -ge 2 ]
355             then
356                 if [ $m -eq 1 ]
357                 then
358                     printf ", "
359                 fi
360                 if [ $counter -eq 0 ]
361                 then
362                     printf "Phone number"
363                 else
364                     n1=$( echo $line | cut -d "," -f3 )
365                     printf $n1
366                 fi
367                 m=1
368             fi
369             mod=$(( $caser % 2 ))
370             if [ $mod -eq 1 ]
371             then
372                 if [ $m -eq 1 ]
373                 then
374                     printf ", "
375                 fi
376                 if [ $counter -eq 0 ]
377                 then
378                     printf "email"
379                 else
380                     n1=$( echo $line | cut -d "," -f4 )
381                     if [ $n1 ]
382                     then
383                         printf $n1
384                     else
385                         printf "Empty"
386                     fi
387                 fi
388             fi
389             counter=$(( $counter + 1 ))
390             m=0
391         done < $f
392     #----- The loop end by the end of the file -----
393     printf "\n"
394     printf "\n"
395     printf "\n"
396 }
```

2.4. The Third Function: Search for a Contact

This function asks the user to enter the info that will be used to search for a user or users who contain the same info depending on one or more fields, if the info data doesn't match any Contact the system will tell the user that the data is invalid and it also makes sure that it's not for the first identifying line.

```
399 #===== The function that will search for a contact =====
400 third () {
401     printf "\n"
402     read -p "Enter an info about that user you are looking for" info
403     (Name, phone number, email, etc...) : " info # Any small info can print any user containing it
404
405     if [[ $info = "*" * ]] # The more than a field case-----
406     then
407         NoWords=$( echo "$info" | wc -w ) # Calculate the number of used fields since they are more than 1
408         info=$( echo "$info" | cut -d" " -f1 )
409         data=$( grep "$info" $f )
410         countSplit=2
411         while [ $countSplit -le $NoWords ] # A loop that goes on all entered informations to search for the users containing them
412         do
413             info=$( echo "$info" | cut -d" " -f$countSplit )
414             data=$( echo "$data" | grep "$info" )
415             countSplit=$(( $countSplit + 1 ))
416         done
417     else # One field case
418         data=$( grep "$info" $f )
419     fi
420     printf "\n"
421     firstLine=$(sed -n 1p $f )
422     if [[ "$data" = "$firstLine" * ]] # In case the searched data was from the first line
423     then
424         data=$( echo "$data" | sed '1d' )
425     fi
426     if [ "$data" ] # To make sure that the data isn't from the first line since its part of the file
427     then
428         echo "$firstLine"
429         echo "$data"
430     else
431         echo "The data you entered isn't valid!"
432     fi
433     printf "\n"
434     printf "\n"
435     printf "\n"
436 }
437
438
```

2.5. The fourth Function: Edit a Contact

This function edits a contact that the user gives info about, if the user gives an info that is for more than a user the system will ask the user to enter the phone number because it can't be repeated for more than one user, if any of the info is not found then the system will tell the user that the given info is invalid, if the info was valid a new menu will be printed to ask the user what he wants to change in that user (name, phone number, email), this function can call one of four changing functions in order to change a user's data.

```

440 ##### The function that will edit a contact #####
441 fourth () {
442     printf "\n"
443     read -p "Enter an info about that user that you want to change"
444     (Name, phone number, email, etc...) : " info # Any small info can print any user containing it
445     z=0
446     if [[ $info = "" ]]
447     then
448         info=$( echo $info | sed 's/ /, /') # As the info of the contact is separated by a coma and a space not just a space
449     fi
450     data=$( grep "$info" $f )
451     if [ "$data" ]
452     then
453         if (( $( grep -c . <<<"$data" ) > 1 )) # Count the number of the contacts with the same info
454         then
455             printf "\n"
456             grep "$info" $f # Show all the contacts with that info so the user can choose exactly which one
457             printf "\n"
458             read -p "The data you entered isn't enough, please enter a phone number: " info # Since more than one contact can't have the same number
459             data=$( grep "$info" $f )
460             if [ -z "$data" ]
461             then
462                 z=1 # To check if the phone number exists
463             fi
464             if [ $z -eq 0 ]
465             then
466                 lines=$( grep -n "$info" $f | cut -d ':' -f1 ) # To check if the data isn't from the first line since its also part of the file
467                 if [ $line -eq 1 ]
468                 then
469                     echo "The data you entered isn't valid!"
470                 else
471                     printf "\n"
472                     echo " $data" # Here is the changing menu
473                     printf "\n"
474                     printf "Changing Menu\n"
475                     printf "===== \n"
476                     printf "1) change First Name\n"
477                     printf "2) change Last Name\n"
478                     printf "3) change, add, or Remove a Phone Number\n"
479                     printf "4) change email\n"
480                     printf "other) to exit\n"
481                     read -p "Enter the choice: " S
482                     case $S
483                     in
484                         1) fnameChange ;; # If S = 1, go the (fnameChange) function
485                         2) lnameChange ;; # If S = 2, go the (lnameChange) function
486                         3) pnumberChange ;; # If S = 3, go the (pnumberChange) function
487                         4) emailChange ;; # If S = 4, go the (emailChange) function
488                         *) echo "leaving the case..."
489                     esac
490                     if
491                     then
492                         echo "this phone number doesn't exist!" # The case of the phone number wasn't for any of the searched data
493                     fi
494                     else
495                         echo "The data you entered isn't valid!" # The case of the first searched data isn't valid
496                     fi
497                     printf "\n"
498                     printf "\n"
499                     printf "\n"
500                 fi
501             }
502 }

```

2.5.1. The First Name Edit Function

This function edits the first name or changes it and checks if the new name only contains alphabetic letters, if not it won't be edited.

```

1 ##### First name change function #####
2 fnameChange () {
3     notChar='^[A-Za-z]+$'
4     read -p "Enter the new First Name: " n
5     if ! [[ $n =~ $notChar ]] # The name must only contain letters
6     then
7         echo "That's not a valid name, so the original name won't be changed"
8     else
9         prev=$( echo $data | cut -d "," -f1 )
10        changedData=$(data//$prev/$n)
11        all=$( sed -e "$line" 'd' $f )
12        echo "$all" > $f
13        echo "$changedData">> $f
14    fi
15 }
16

```

2.5.2. The Last Name Edit Function

This function edits the second name and adds one if there isn't and checks if it only contains alphabetic letters if not it won't be edited.

```

17 ##### Last name change function #####
18 lnameChange () {
19     notChar='^[A-Za-z]+$'
20     read -p "Enter the new Last Name: " n
21     if ! [[ $n =~ $notChar ]] # The name must only contain letters
22     then
23         echo "That's not a valid name, so the original name won't be changed"
24     else
25         changedData=$( echo "$data" | cut -d "," -f1 )
26         changedData+=", $n"
27         changedData+=", "
28         phonenumber=$( echo "$data" | cut -d "," -f3 )
29         changedData+="$phonenumber"
30         email=$( echo "$data" | cut -d "," -f4 )
31         changedData+=", "
32         changedData+="$email"
33         all=$( sed -e "$line" 'd' $f )
34         echo "$all" > $f
35         echo "$changedData">> $f
36     fi
37 }
38

```


2.5.3. The Phone Number Edit Function

This function gives the user three options for the searched contact (add a new number, edit a number, delete a number) if the user is entering a new number or editing one then the system checks if the new number only contains numbers and that it is from 9-10 digits and if the user is editing or deleting a number the system checks if the user has the old changed/deleted number, if the contact has only one number it can't be deleted because each contact must at least have one number.

```

39 ##### Phone numbe change function #####
40 pnumberChange () {
41     printf "\n"
42     read -p "Enter 1 if you want to add a new phone number,
43     2 : to change a phone number,
44     3 to delete a phone number,
45     other to leave
46     =====
47     chooser:" selector # A menu to choose what to do with the given phone number/s
48     prev=$( echo "$data" | cut -d "," -f3) # To count the number of Phone Numbers
49     count=$( {prev//[!0-9]} )
50     nop=$((count))
51     nop=$(( $nop + 1 ))
52     set=0
53     old="" # old should store the value of the old phone number if its being deleted or edited
54     mode=1 # mode=1 means that chooser entered a valid phone number to edit/delete
55
56     ##### Finding the old number to replace/ delete #####
57     if [ $selector = "2" -o $selector = "3" ]
58     then
59         if [ $nop -gt 1 ] #----- The case of user has more than one number -----
60         then
61             read -p "Enter the old Phone Number (Since there is more than a number): " old # check if the number exists
62             if [ "$prev" == "$old" ]
63             then
64                 re='^[0-9]+$'
65                 if ! [[ $old =~ $re ]]
66                 then # check if it is only numbers
67                     echo "The Number you entered isn't valid (contains other things) so it will not be added/changed"
68                     mode=0
69                 else
70                     checking=$( echo -n $old | wc -c )
71                     if [ $checking -ge 9 -a $checking -le 10 ] # check if its a full number (9-10 numbers)
72                     then
73                         mode=1
74                     else
75                         echo "You didn't enter the real full number!" # The case that the entered number isn't full
76                         mode=0
77                     fi
78                 fi
79             else
80                 echo "This persons doesn't own this number!" # The entered number isn't among the available ones
81                 mode=0
82             fi
83         else #----- The case of the user has only one number -----
84             if [ $selector = "3" ] # For the delete case-----
85             then
86                 mode=0 # Since the user has to have a number, if he has only one it can't be deleted
87                 echo "This user has only one phone number, add a new number in order to delete this one"
88             else # For thr edit case-----
89                 old=$( echo "$prev" | sed 's/ /g' ) # The old number simply is the only available number
90                 fi
91             fi
92         fi
93     fi
94     ##### Setting a new number to add/ replace #####
95     if [ $selector = "1" -o $selector = "2" ]
96     then
97         if [ $mode -eq 1 ] # To check for no wrong cases if there is an old number being replaced
98         then
99             read -p "Enter the new Phone Number: " n # The user will enter a new number
100             re='^[0-9]+$' # Checking if the new number is only consists of numbers
101             if ! [[ $n =~ $re ]]
102             then
103                 echo "The Number you entered isn't valid (contains other things) so it will not be added/changed"
104             else
105                 checking=$( echo -n $n | wc -c ) # Checking if the new number has a wanted lenght
106                 if [ $checking -ge 9 -a $checking -le 10 ]
107                 then
108                     alreadyExists=$( grep "$n" $f ) # Check if any user already has this number
109                     if [ "$alreadyExists" ]
110                     then
111                         echo "This number already exists"
112                     else
113                         if [ $selector = "1" ] #----- The case of adding new number -----
114                         then
115                             new="$prev"
116                             new+=", $n"
117                             changedData=$(data//"$prev"/"$new")
118                         else #----- The case of replacing a number -----
119                             changedData=$(data//"$old"/"$n")
120                             fi
121                             all=$( sed -e "$line'd' $f ) #----- Replacing the old data with the new -----
122                             echo $all > $f
123                             echo "$changedData">> $f
124                         fi
125                     else
126                         echo "The Number you entered isn't valid (size issue), so it will not be added/changed."
127                     fi
128                 fi
129             fi
130         fi
131     fi
132     ##### Deleting the searched number #####
133     elif [ $selector = "3" -a $mode -eq 1 ] # If mode equals 0 that means either number is not found or user has one number, on both cases it will refuse to delete
134     then
135         echo "Sold this number is being deleted!"
136         changedData=$(data//"$old"/)
137         changedData=$( echo "$changedData" | sed -r 's/;/;/;' ) # replacing the semicolon 3 cases depending on the location of the deleted number
138         changedData=$( echo "$changedData" | sed -r 's/;/;/g' )
139         changedData=$( echo "$changedData" | sed -r 's/;/;/g' )
140         all=$( sed -e "$line'd' $f )
141         echo $all > $f
142         echo "$changedData">> $f
143     else
144         echo "You didn't enter a valid chice!"
145     fi
146 }

```

2.5.4. The Email Change Function

This function changes the email or adds a one if the contact didn't previously have one and checks that it contains only one @.

```
146 ##### Email change function #####
147 emailChange () {
148     read -p "Enter the new Email: " n
149     at="@"
150     if [[ "$n" == *$at* ]]
151     then
152         C=$(n//[^\@])
153         CO=$(#C)
154         if [ $CO -eq 1 ]
155         then
156             prev=$( echo "$data" | cut -d "," -f4 )
157             if [ "$prev" ]
158             then
159                 ns="$n"
160                 changedData=$(data//$prev/"$ns")
161             else
162                 changedData=$data
163                 changedData+=" $n"
164             fi
165             all=$( sed -e "$line" d' $f )
166             echo "$all" > $f
167             echo "$changedData" >> $f
168         else
169             echo "The email you entered isn't valid!"
170         fi
171     else
172         echo "The email you entered isn't valid!"
173     fi
174 }
175
```

2.6. The Fifth Function: Deleting a contact

This function searches for a contact based on given info, if the info is already there then the system will ask the user to enter the phone number and then it will print the contact before deleting it, then it deletes it.

```
504 ##### The function that will delete a contact #####
505 fifth () {
506     printf "\n"
507     read -p "Enter an info about that user that you want to Delete" info
508     (Name, phone number, email, etc...) : " info
509     z=0
510     if [[ $info == *" " ]]
511     then
512         info=$( echo $info | sed 's/ /, /' )
513         # As the info of the contact is separated by a coma and a space not just a space
514         data=$( grep "$info" $f )
515         if [ "$data" ]
516         then
517             if (( $( grep -c . <<<"$data" ) > 1 ))
518             then
519                 # Count the number of the contacts with the same info
520                 printf "\n"
521                 grep "$info" $f
522                 printf "\n"
523                 read -p "The data you entered isn't enough, please enter a phone number: " info
524                 data=$( grep "$info" $f )
525                 if [ -z "$data" ]
526                 then
527                     z=1
528                     # To check if the phone number exists
529                 fi
530             fi
531             if [ $z -eq 0 ]
532             then
533                 lines=$( grep -n "$info" $f | cut -d ":" -f1 )
534                 if [ "$line" -eq 1 ]
535                 then
536                     echo "The data you entered isn't valid!"
537                 else
538                     printf "\n"
539                     echo " $data"
540                     # Print the contact before it's deleted
541                     echo " This user has been deleted!"
542                     # save all data without the deleted line then replace it in the file
543                     all=$( sed -e "$line" d' $f )
544                     echo "$all" > $f
545                 fi
546             fi
547             echo "this phone number doesn't exist!"
548             # The case of the phone number wasn't for any of the searched data
549         else
550             echo "The data you entered isn't valid!"
551             # The case of the first searched data isn't valid
552         fi
553     fi
554     printf "\n"
555     printf "\n"
556     printf "\n"
557 }
558
```

3. Testing Examples

There are 2 testing examples of a Contact.txt file that contains the following contacts:

```
1 First name, Last name, Phone numbers, email
2 Ahmaide, Awawdah, 0592124009;0595873953, ahmaidy@gmail.com
3 Michael, , 0561710003,m@mail.com
4 Trevor, Phillips, 0591713333;0567713490,
5 Justin, , 0590013434,
6 Malik, Ahmad, 0591713223;0892323111, MalikA@mail.com
7 Mohammad, Khalil, 0561311313, mohammadK@yahoo.com
8 Margot, , 0599237676;0561119998, mar@aol.com
9 James, Bond, 0001213998,
```

3.1. Test Example 1

```
ahmaide@ahmaide-VirtualBox:~/Desktop$ ./project1.txt
Enter File Name: Contact.txt
**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 1

Enter first name: Ahmad01
That's not a valid name, try again
Enter first name: Ahmad
Do you want to enter a last name (if yes enter 1, other if not): 1
Enter last name: Waleed
Enter the number of phone numbers: 2
Enter phone 1 : 059
The Number you entered isn't valid (size issue), Please try again.
Enter phone 1 : 0591419990
Enter phone 2 : 056asa1212
The Number you entered isn't valid (contains other things), Please try again.
Enter phone 2 : 0567780110
Do you want to enter a email (if yes enter 1, other if not): 1
Enter Email: ahmaid1mail
The email you entered isn't valid!
Enter Email: ahmad1g@mail
The email you entered isn't valid!
Enter Email: ahmad1@gmail.com

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 2

If you want the first name to be printed enter 1, enter others to not: 1
If you want the last name to be printed enter 1, enter others to not: 1
If you want the phone numbers to be printed enter 1, enter others to not: 1
If you want the email to be printed enter 1, enter others to not: 1

First name, Last name, Phone number, email
Ahmaide, Awawdah, 0592124009;0595873953, ahmaidy@gmail.com
Michael, Empty, 0561710003, m@mail.com
Trevor, Phillips, 0591713333;0567713490, Empty
Justin, Empty, 0590013434, Empty
Malik, Ahmad, 0591713223;0892323111, MalikA@mail.com
Mohammad, Khalil, 0561311313, mohammadK@yahoo.com
Margot, Empty, 0599237676;0561119998, mar@aol.com
James, Bond, 0001213998, Empty
Ahmad, Waleed, 0591419990;0567780110, ahmad1@gmail.com
```

```

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 3

Enter an info about that user you are looking for
(Name, phone number, email, etc...) : Ah 059

First name, Last name, Phone numbers, email
Ahmaide, Awawdah, 0592124009;0595873953, ahmaidy@gmail.com
Malik, Ahmad, 0591713223;0892323111, MalikA@mail.com
Ahmad, Waleed, 0591419990;0567780110, ahmad1@gmail.com

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 4

Enter an info about that user that you want to change
(Name, phone number, email, etc...) : Ahm

Ahmaide, Awawdah, 0592124009;0595873953, ahmaidy@gmail.com
Malik, Ahmad, 0591713223;0892323111, MalikA@mail.com
Ahmad, Waleed, 0591419990;0567780110, ahmad1@gmail.com

The data you entered isn't enough, please enter a phone number: 0592124009

Ahmaide, Awawdah, 0592124009;0595873953, ahmaidy@gmail.com

Changing Menu
=====
1) Change First Name
2) Change Last Name
3) change, add, or Remove a Phone Number
4) Change email
other) to exit
Enter the choice: 2
Enter the new Last Name: Taha

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 5

Enter an info about that user that you want to Delete
(Name, phone number, email, etc...) : Waleed

Ahmad, Waleed, 0591419990;0567780110, ahmad1@gmail.com
This user has been deleted!

```

3.2. Test Example 2

```
ahmad@ahmad-VirtualBox:~/Desktop$ ./project1.txt
Enter File Name: Contact.txt
**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 1

Enter first name: Mike
Do you want to enter a last name (if yes enter 1, other if not): 0
Enter the number of phone numbers: 1
Enter phone 1 : 000121999
Do you want to enter a an email (if yes enter 1, other if not): 1
Enter Email: mike@yahoo.com

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 2

If you want the first name to be printed enter 1, enter others to not: 1
If you want the last name to be printed enter 1, enter others to not: 1
If you want the phone numbers to be printed enter 1, enter others to not: 0
If you want the email to be printed enter 1, enter others to not: 1

First name, Last name, email
Michael, Empty, m@mail.com
Trevor, Phillips, Empty
Justin, Empty, Empty
Malik, Ahmad, MalikA@mail.com
Mohammad, Khalil, mohammadK@yahoo.com
Margot, Empty, mar@aol.com
James, Bond, Empty
Ahmaide, Taha, ahmaidy@gmail.com
Mike, Empty, mike@yahoo.com

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 4

Enter an info about that user that you want to change
(Name, phone number, email, etc...) : Malik
Malik, Ahmad, 0591713223;0892323111, MalikA@mail.com

Changing Menu
=====
1) Change First Name
2) Change Last Name
3) change, add, or Remove a Phone Number
4) Change email
other) to exit
Enter the choice: 3

Enter 1 if you want to add a new phone number,
2 : to change a phone number,
3 to delete a phone number,
other to leave
=====
chooser:2
Enter the old Phone Number (Since there is more than a number): 0591713223
Enter the new Phone Number: 0599131880

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 2

If you want the first name to be printed enter 1, enter others to not: 1
If you want the last name to be printed enter 1, enter others to not: 1
If you want the phone numbers to be printed enter 1, enter others to not: 1
If you want the email to be printed enter 1, enter others to not: 1

First name, Last name, Phone number, email
Michael, Empty, 0561710003, m@mail.com
Trevor, Phillips, 0591713333;0567713490, Empty
Justin, Empty, 0590013434, Empty
Mohammad, Khalil, 0561311313, mohammadK@yahoo.com
Margot, Empty, 0599237676;0561119998, mar@aol.com
James, Bond, 0001213998, Empty
Ahmaide, Taha, 0592124009;0595873953, ahmaidy@gmail.com
Mike, Empty, 0000121999, mike@yahoo.com
Malik, Ahmad, 0599131880;0892323111, MalikA@mail.com
```

```

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 5

Enter an info about that user that you want to Delete
(Name, phone number, email, etc...) : Marg

Margot, , 0599237676;0561119998, mar@aol.com
This user has been deleted!

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 2

If you want the first name to be printed enter 1, enter others to not: 1
If you want the last name to be printed enter 1, enter others to not: 0
If you want the phone numbers to be printed enter 1, enter others to not: 0
If you want the email to be printed enter 1, enter others to not: 1

First name, email
Michael, m@mail.com
Trevor, Empty
Justin, Empty
Mohammad, mohammadK@yahoo.com
James, Empty
Ahmaide, ahmaidy@gmail.com
Mike, mike@yahoo.com
Malik, MalikA@mail.com

**** Welcome to Contact Managment System ****

Main Menu
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
Enter the choice: 0
ahmaide@ahmaide-VirtualBox:~/Desktop$

```

4. Conclusion

In Conclusion it was noticed that shell script can be very useful in Linux operating system for file read and checking and searching and this project can be a good example of the big use of shell scripting and how to deal with things (files, folders, programs ...etc) on Linux using simple commands.

5. Appendix

#First Name, Last Name, ID, Section

#Ahmaide, Awawdah, 1190823, 1

#Khalid, Addabe, 1190507, 1

#===== First name change function =====

```
fnameChange () {
    notChar='^[A-Za-z]+$'
    read -p "    Enter the new First Name: " n
    if ! [[ $n =~ $notChar ]]
        then
            echo "    That's not a valid name, so the original name won't be changed"
        else
            prev=$( echo $data | cut -d "," -f1)
            changedData=${data//$prev/$n}
            all=$( sed -e "$line""d" $f )
            echo "$all" > $f
            echo "$changedData">> $f
        fi
}
```

#===== Last name change function =====

```
LnameChange () {
    notChar='^[A-Za-z]+$'
    read -p "    Enter the new Last Name: " n
    if ! [[ $n =~ $notChar ]]
        then
            echo "    That's not a valid name, so the original name won't be changed"
        else
            changedData=$( echo "$data" | cut -d "," -f1)
            changedData+=", $n"
            changedData+=", "
            phonenumber=$( echo "$data" | cut -d "," -f3)
            changedData+="$phonenumber"
            email=$( echo "$data" | cut -d "," -f4)
            changedData+=", "
            changedData+="$email"
            all=$( sed -e "$line""d" $f )
            echo "$all" > $f
            echo "$changedData">> $f
        fi
}
```

#===== Phone numbe change function =====

```
PnumberChange () {
    printf "\n"
    read -p "    Enter 1 if you want to add a new phone number,
2 : to change a phone number,
3 to delete a phone number,
other to leave
=====
chooser:" selector
prev=$( echo "$data" | cut -d "," -f3)
count="{prev//[^;]}"
nop="{#count}"
nop=$(( $nop + 1 ))
set=0
# A menu to choose what to do with the given phone number/s
# To count the number of Phone Numbers
```



```

old="" # old should store the value of the old phone number if its being deleted or edited
mode=1 # mode=1 means that chooser entered a valid phone number to edit/delete

#----- Finding the old number to replace/ delete -----
if [ $selector = "2" -o $selector = "3" ]
then
    if [ $nop -gt 1 ]
    #----- The case of user has more than one number -----
    then
        read -p " Enter the old Phone Number (Since there is more than a number): " old
        if [[ "$prev" == "$old" ]] # check if the number exists
        then
            re='^[0-9]+$'
            if ! [[ $old =~ $re ]] # check if it is only numbers
            then
                echo " The Number you entered isn't valid (contains other things) so it will not be
added/changed"
                mode=0
            else
                checking=$( echo -n $old | wc -c )
                if [ $checking -ge 9 -a $checking -le 10 ] # check if its a full number (9-10 numbers)
                then
                    mode=1
                else
                    echo " You didn't enter the real full number!"
                # The case that the entered number isn't full
                mode=0
            fi
        fi
    else
        echo " This persons doesn't own this number!"
        # The entered number isn't among the available ones
        mode=0
    fi
else
    #----- The case of the user has only one number -----
    if [ $selector = "3" ] # For the delete case-----
    then
        mode=0
        # Since the user has to have a number, if he has only one it can't be deleted
        echo " This user has only one phone number, add a new number in order to delete this one"
        # For thr edit case-----
    else
        old=$( echo "$prev" | sed 's/ //g' ) # The old number simply is the only available number
    fi
fi
#----- Setting a new number to add/ replace -----
if [ $selector = "1" -o $selector = "2" ]
then
    if [ $mode -eq 1 ] # To check for no wrong cases if there is an old number being replaced
    then
        read -p " Enter the new Phone Number: " n # The user will enter a new number
        re='^[0-9]+$' # Checking if the new number is only consists of numbers
        if ! [[ $n =~ $re ]]
        then
            echo " The Number you entered isn't valid (contains other things) so it will not be
added/changed"
        else
            checking=$( echo -n $n | wc -c ) # Checking if the new
number has a wanted lenght
            if [ $checking -ge 9 -a $checking -le 10 ]

```

```

                                then
already has this number                                alreadyExists=$( grep "$n" $f )                                # Check if any user
                                if [ "$alreadyExists" ]
                                then
                                        echo "    This number already exists"
                                else
                                        if [ $selector = "1" ]                                #----- The case of
                                        then
                                                new="$prev"
                                                new+=";$n"
                                                changedData=${data//"$prev"/"$new"}
                                        else                                #-----
                                                changedData=${data//"$old"/"$n"}
                                        fi
                                        all=$( sed -e "$line"d' $f )                                #----- Replacing the
                                        echo "$all" > $f
                                        echo "$changedData">> $f
                                fi
                                else
added/changed."                                echo "    The Number you entered isn't valid (size issue), so it will not be
                                fi
                                fi
                                #----- Deleting the searched number -----
                                elif [ $selector = "3" -a $mode -eq 1 ]
                                then                                # If mode equals 0 that means either number is not found or user
has one number, on both cases it will refuse to delete                                echo "    $old this number is being deleted!"
                                changedData=${data//$old/}
                                changedData=$( echo "$changedData" | sed -r 's/;;/;' )                                # replacing the semicolon 3 cases
depending on the location of the deleted number                                changedData=$( echo "$changedData" | sed -r 's/;/;/g' )
                                changedData=$( echo "$changedData" | sed -r 's/;;/;/g' )
                                all=$( sed -e "$line"d' $f )
                                echo "$all" > $f
                                echo "$changedData">> $f
                                else
                                echo "    You didn't enter a valid chice!"
                                fi
}

#===== Email change function
=====
emailChange () {
    read -p "    Enter the new Email: " n
    at="@ "
    if [[ "$n" == *"$at"* ]]                                # To check if the email has a @
    then
        C="{n//[^\@]}"
        CO="{#{C}"
        if [ $CO -eq 1 ]                                # To check that the email has only one @
        then
            prev=$( echo "$data" | cut -d " " -f4 )
            if [ "$prev" ]
            then
                ns=" $n"

```

```

                                changedData=${data//"$prev"/"$sns"}
                                else
                                changedData=$data
                                changedData+=" $n"
                                fi
                                all=$( sed -e "$line"d' $f )
                                echo "$all" > $f
                                echo "$changedData">> $f
                                else
                                echo "    The email you entered isn't valid!"
                                fi
                                else
                                echo "    The email you entered isn't valid!"
                                fi
                                fi
                                }

#===== The function that will add a new contact
=====
first () {
    printf "\n"
    #----- Adding First name -----
    notChar='^[A-Za-z]+$'
    chooser=0
    while [ $chooser -eq 0 ]
    do
        read -p "    Enter first name: " fn          # The user will enter the new first name
        if ! [[ $fn =~ $notChar ]]                  # The name must only contain letters
        then
            echo "    That's not a valid name, try again"
        else
            chooser=1
        fi
    done
    allInfo=$fn                                     # allInfo is a string to contain all the user's info
    allInfo+=", "                                   # Add a coma to separate the first name from the
second info
    #----- Adding a last name by the user's choice -----
    read -p "    Do you want to enter a last name (if yes enter 1, other if not): " chooser
    if [ $chooser = "1" ]
    then
        notChar='^[A-Za-z]+$'
        chooser=0
        while [ $chooser -eq 0 ]
        do
            read -p "    Enter last name: " ln          # The user will enter the new last name
            if ! [[ $ln =~ $notChar ]]                  # The name must only contain
letters
            then
                echo "    That's not a valid name, try again"
            else
                chooser=1
            fi
        done
        allInfo+=$ln
    fi
    allInfo+=", "
    #----- Adding phone number/s -----
    read -p "    Enter the number of phone numbers: " nop
    co=1                                             # co is the counter of phone
numbers

```

```

allp=""
while [ $co -le $nop ]                                # A counter to get all the phone numbers
do
    read -p "    Enter phone $co : " pn                # The user enters the phone number and co is the
order of the phone number
    re='^[0-9]+$'
    if ! [[ $pn =~ $re ]]                                # re is to make sure that the entered value
only contains numbers
    then
        echo "    The Number you entered isn't valid (contains other things), Please try again."    # If the
user entered stuff other than numbers
    else
        checking=$( echo -n $pn | wc -c )                # checking contains the number of charcters in the
numbers which should be 9-10
        if [ $checking -ge 9 -a $checking -le 10 ]
        then
            alreadyExists=$( grep "$pn" $f )                # alreadyExists is to check if someone else already has
this phone number
            if [ "$alreadyExists" ]
            then
                echo "    This number already exists"
                if [ $nop -eq 1 ]                            # In this case the system won't let the user enter a new
number instead unless it was the only number
                then
                    echo "    you need to enter a new phone number!"
                else
                    nop=$(( $nop - 1 ))                    # So the number of the entire numbers will
decrease if the user entered an exisiting number
                fi
            else                                            # The case of the number is a
new number
                allp+=$pn
                if [ $co -ne $nop ]                        # This if statment is to check if there is still
numbers to add in order to add semicolon
                then
                    allp+=";"
                fi
                co=$(( $co + 1 ))
            fi
        else                                            # The case of that the length of
the number is not between 9-10
            echo "    The Number you entered isn't valid (size issue), Please try again."
        fi
    fi
done
allInfo+=$allp
allInfo+=", "
#----- Adding an email by the user's chice -----
read -p "    Do you want to enter a an email (if yes enter 1, other if not): " chooser
if [ $chooser = "1" ]
then
    chooser=0
    at="@"
    while [ $chooser -eq 0 ]                            # This loop will keep going until the user enters an
email with one @
    do
        read -p "    Enter Email: " e                    # e is the string of the email
        if [[ "$e" == "$at"* ]]                            # Here to check if @ is included
        then
            C="{e//[!^@]}"

```

```

        C0="{#{C}"
        if [ $C0 -eq 1 ]                                # Here to check that only one @ is included
        then
            chooser=1
        else
            echo "    The email you entered isn't valid!"
        fi
    else
        echo "    The email you entered isn't valid!"
    fi
done
allInfo+=$e
fi
#-----Add all the info in a new line to the contact file -----
echo $allInfo >> $f
printf "\n"
printf "\n"
printf "\n"
}

#===== The function that will print the contact data
=====
second () {
    #-----# Here the user selects what to print and the if statement works in binary bits -----
    -----
    printf "\n"
    caser=0
    read -p "    If you want the first name to be printed enter 1, enter others to not: " chooser
    if [ $chooser = "1" ]
    then
        caser=$(( $caser + 8 ))
    fi
    read -p "    If you want the last name to be printed enter 1, enter others to not: " chooser
    if [ $chooser = "1" ]
    then
        caser=$(( $caser + 4 ))
    fi
    read -p "    If you want the phone numbers to be printed enter 1, enter others to not: " chooser
    if [ $chooser = "1" ]
    then
        caser=$(( $caser + 2 ))
    fi
    read -p "    If you want the email to be printed enter 1, enter others to not: " chooser
    if [ $chooser = "1" ]
    then
        caser=$(( $caser + 1 ))
    fi
    m=0
    printf "\n"
    counter=0
    #-----A loop to go around all the lines -----
    while read line;
    do
        printf "\n"
        printf "    "
        if [ $caser -ge 8 ]                                # If the user asked to print the first name the caser will
        be higher than 8
        then
            if [ $counter -eq 0 ]                            # Counter=0 means that its reading the first line ( first
            name, last name, ...., etc)

```

```

        then
            printf "First name"
        else
            n1=$( echo $line | cut -d "," -f1) # Take the first seperated part by comas
            printf $n1
        fi
        m=1 # m=1 means that there is a previous data
    printed in order to add a coma
        fi
        mod=$(( $caser % 8 ))
        if [ $mod -ge 4 ] # If the user asked to print the last name the caser
            then
                if [ $m -eq 1 ] # To print coma if a previous data was printed
                then
                    printf ", "
                fi
                if [ $counter -eq 0 ]
                then
                    printf "Last name"
                else
                    n1=$( echo $line | cut -d "," -f2) # Take the second seperated part by comas
                    if [ $n1 ]
                    then
                        printf $n1
                    else
                        printf "Empty" # If the user has no last name then print
                    fi
                fi
            fi
            m=1
        fi
        mod=$(( $caser % 4 ))
        if [ $mod -ge 2 ] # If the user asked to print the phone number the
            then
                if [ $m -eq 1 ]
                then
                    printf ", " # To print coma if a previous data was printed
                fi
                if [ $counter -eq 0 ]
                then
                    printf "Phone number"
                else
                    n1=$( echo $line | cut -d "," -f3) # Take the third seperated part by comas
                    printf $n1
                fi
            fi
            m=1
        fi
        mod=$(( $caser % 2 ))
        if [ $mod -eq 1 ] # If the user asked to print the email the caser would
            then
                if [ $m -eq 1 ]
                then
                    printf ", " # To print coma if a previous data was printed
                fi
                if [ $counter -eq 0 ]
                then
                    printf "email"
                else
                    printf " "
                fi
            fi
        fi
    fi
fi

```

```

        n1=$( echo $line | cut -d "," -f4) # Take the fourth seperated part by comas
        if [ $n1 ]
        then
            printf $n1                # If the user has no email then print "Empty"
        else
            printf "Empty"
        fi
    fi
fi
counter=$(( $counter + 1 ))
m=0
done < $f
#-----The loop end by the end of the file -----
printf "\n"
printf "\n"
printf "\n"
}

#===== The function that will search for a contact
=====
third () {
    printf "\n"
    read -p "    Enter an info about that user you are looking for
(Name, phone number, email, etc...) : " info
    containing it                                     # Any small info can print any user
    if [[ $info = "*" "*" ]]
    then
        NofWords=$( echo "$info" | wc -w )
        infoS=$( echo "$info" | cut -d " " -f1 )
        data=$( grep "$infoS" $f )
        countSplit=2
        while [ $countSplit -le $NofWords ]
        do
            infoS=$( echo "$info" | cut -d " " -f$countSplit )
            data=$( echo "$data" | grep "$infoS" )
            countSplit=$(( $countSplit + 1 ))
        done
    else
        data=$( grep "$info" $f )
    fi
    printf "\n"
    firstLine=$(sed -n 1p $f )
    if [[ "$data" = "$firstLine"* ]]
    then
        data=$( echo "$data" | sed '1d' )
    fi
    if [ "$data" ]
    then
        # To make sure that the data isn't from the first line since its part
        of the file
        echo "$firstLine"
        echo "$data"

    else
        echo "    The data you entered isn't valid!"
    fi
    printf "\n"
    printf "\n"
    printf "\n"
}

```

```

#===== The function that will edit a contact
=====
fourth () {
    printf "\n"
    read -p "    Enter an info about that user that you want to change
(Name, phone number, email, etc...) : " info
    # Any small info can print any user
    containing it
    z=0
    if [[ $info = "*" "*" ]]
    then
        info=$( echo $info | sed 's/ /, /')
        # As the info of the contact is separated by
        a coma and a space not just a space
    fi
    data=$( grep "$info" $f )
    if [ "$data" ]
    then
        if (( $( grep -c . <<<"$data" ) > 1 ))
        # Count the number of the contacts with
        the same info
        then
            printf "\n"
            grep "$info" $f
            # Show all the
            contacts with that info so the user can choose exactly which one
            printf "\n"
            read -p "    The data you entered isn't enough, please enter a phone number: " info
            # Since more than
            one contact can't have the same number
            data=$( grep "$info" $f )
            if [ -z "$data" ]
            then
                z=1
                # To check if the
                phone number exists
            fi
        fi
        if [ $z -eq 0 ]
        then
            line=$( grep -n "$info" $f | cut -d":" -f1 )
            # To check if the data isn't from the first line
            since its also part of the file
            if [ $line -eq 1 ]
            then
                echo "    The data you entered isn't valid!"
            else
                printf "\n"
                echo "    $data"
                # Here is the changing menu
                printf "\n"
                printf "        Changing Menu\n"
                printf "        =====\n"
                printf "        1) Change First Name\n"
                printf "        2) Change Last Name\n"
                printf "        3) change, add, or Remove a Phone Number\n"
                printf "        4) Change email\n"
                printf "        other) to exit\n"
                read -p "    Enter the choice: " S
                case $S
                in
                    1) fnameChange ;;
                    2) lnameChange ;;
                    3) pnumberChange ;;
                    4) emailChange ;;
                    *) echo "    leaving the case..."
                esac
            fi
        fi
    fi
}

```



```

else
    echo "    this phone number doesn't exist!"                # The case of the phone number wasn't for
any of the searched data
fi
else
    echo "    The data you entered isn't valid!"                # The case of the first searched data isn't valid
fi
printf "\n"
printf "\n"
printf "\n"
}

#===== The function that will delete a contact
=====
fifth (){
    printf "\n"
    read -p "    Enter an info about that user that you want to Delete
(Name, phone number, email, etc...) : " info                # Any small info can print any
user containing it
    z=0
    if [[ $info = "*" "*" ]]
    then
        info=$( echo $info | sed 's/ /, /')                # As the info of the contact is seperated by
a coma and a space not just a space
    fi
    data=$( grep "$info" $f )
    if [ "$data" ]
    then
        if (( $( grep -c . <<<"$data" ) > 1 ))                # Count the number of the
contacts with the same info
        then
            printf "\n"
            grep "$info" $f                # Show all the contacts with that
info so the user can choose exactly which one
            printf "\n"
            read -p "    The data you entered isn't enough, please enter a phone number: " info    # Since more than
one contact can't have the same number
            data=$( grep "$info" $f )
            if [ -z "$data" ]
            then
                z=1                # To check
            fi
            if the phone number exists
            fi
            if [ $z -eq 0 ]
            then
                line=$( grep -n "$info" $f | cut -d":" -f1 )
                if [ $line -eq 1 ]                # To check if the data
isn't from the first line since its also part of the file
                then
                    echo "    The data you entered isn't valid!"
                else
                    printf "\n"
                    echo "    $data"                # Print the contact
before it's deleted
                    echo "    This user has been deleted!"
                    all=$( sed -e "$line""d" $f )                # save all data
without the deleted line then replace it in the file
                    echo "$all" > $f
                fi
            fi
        fi
    fi
}

```

```

                else
                    echo "    this phone number doesn't exist!"                # The case of the phone number
wasn't for any of the searched data
                fi
            else
                echo "    The data you entered isn't valid!"                # The case of the first searched data isn't
valid
            fi
            printf "\n"
            printf "\n"
            printf "\n"
        }

#===== Main menu that first appears on the user screen
=====

read -p "Enter File Name: " f                # f is a variable of the file that will contain the data of users

if [ -e $f ]                # Only do the following if the file exists
then
    select=0                # The user menu at first
    printf "**** Welcome to Contact Managment System ****\n\n"
    printf "    Main Menu\n"
    printf "    =====\n"
    printf "    [1] Add a new Contact\n"
    printf "    [2] List all Contexts\n"
    printf "    [3] Search for Contact\n"
    printf "    [4] Edit a Contact\n"
    printf "    [5] Delete a Contact\n"
    printf "    [0] Exit\n"
    read -p "    Enter the choice: " select                # Here the user enters his selection
    m=(wc -l "$f")
    while [ $select -ge 1 -a $select -le 5 ]                # a loop so the menu keeps appearing until the user asks it to stop
    do
        case "$select"
        in
            1) first ;;                # If select = 1, go the (first) function
            2) second ;;                # If select = 2, go the (second) function
            3) third ;;                # If select = 3, go the (third) function
            4) fourth ;;                # If select = 4, go the (fourth) function
            5) fifth ;;                # If select = 5, go the (fifth) function
            *) break ;;                # In other cases, end the loop
        esac
        printf "**** Welcome to Contact Managment System ****\n\n"
        printf "    Main Menu\n"
        printf "    =====\n"
        printf "    [1] Add a new Contact\n"
        printf "    [2] List all Contexts\n"
        printf "    [3] Search for Contact\n"
        printf "    [4] Edit a Contact\n"
        printf "    [5] Delete a Contact\n"
        printf "    [0] Exit\n"
        read -p "    Enter the choice: " select
    done
else
    echo "Error the file you entered doesn't exist"                # for the file not found case
fi

```