Faculty of Engineering & Technology – Electrical & Computer Engineering Department

First Semester 2021 – 2022

Linux Laboratory – ENCS3130.

Shell Script Project (Project 1)

Name: Ahmaide Al-Awawdah.

ID: 1190823

Name: Khalid Addabe

ID: 1190507

Section: 1

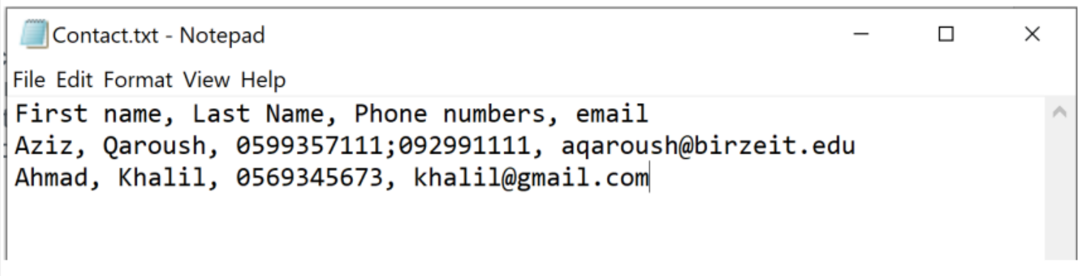
Instructor: Dr. Mohammad Jubran

TA: Eng.Katy Sadi

Date: 7th – 29th November 2021

1. Abstract

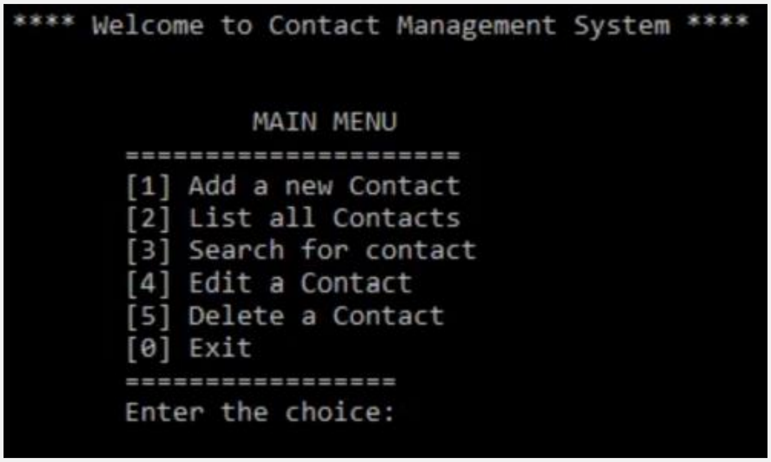
This is project which the code is written in shell script to do operations on a contact file that contains user's data (first name, last name, phone numbers, and email) as shown in the figure below.



```
First name, Last Name, Phone numbers, email
Aziz, Qaroush, 0599357111;092991111, aqaroush@birzeit.edu
Ahmad, Khalil, 0569345673, khalil@gmail.com
```

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.



```
**** Welcome to Contact Management 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:
```

Figure 2: Operations Menu

❖Table of Content

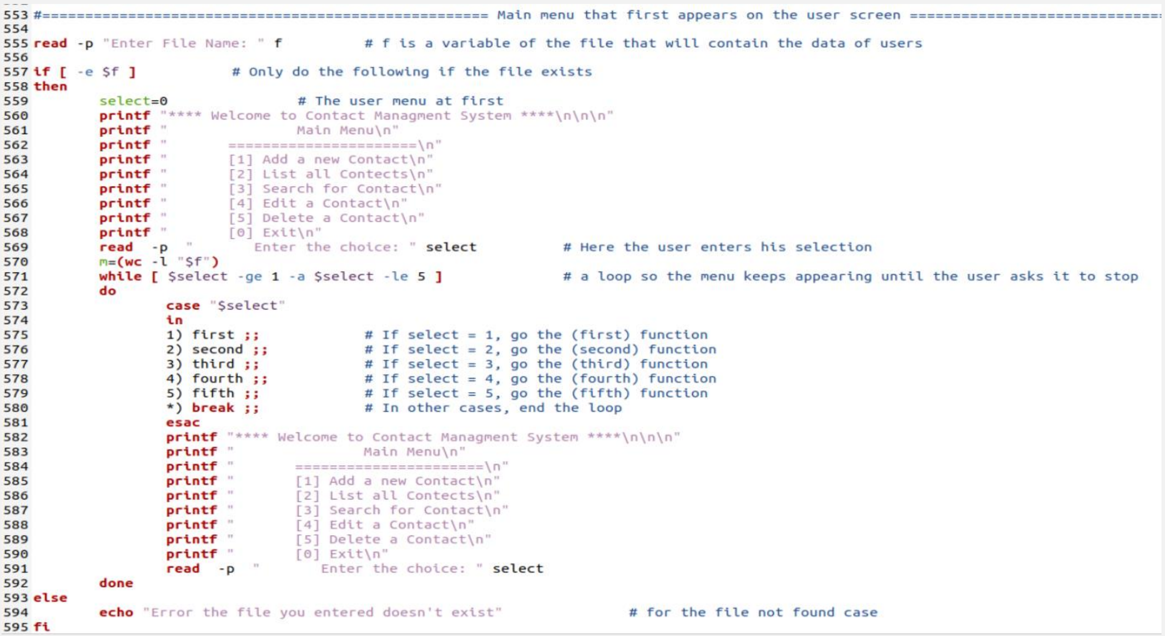
2. The Code

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

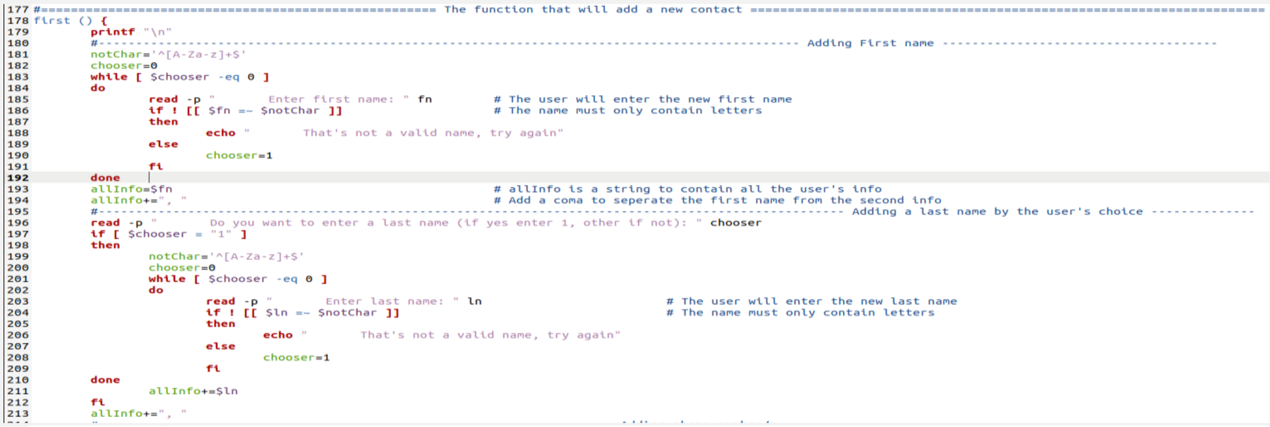
The part where the code starts compiling from where it calls each function only at it's case depending on the user's choice after displaying the menu, but first it needs to ask the user for the file name to see if it is available.



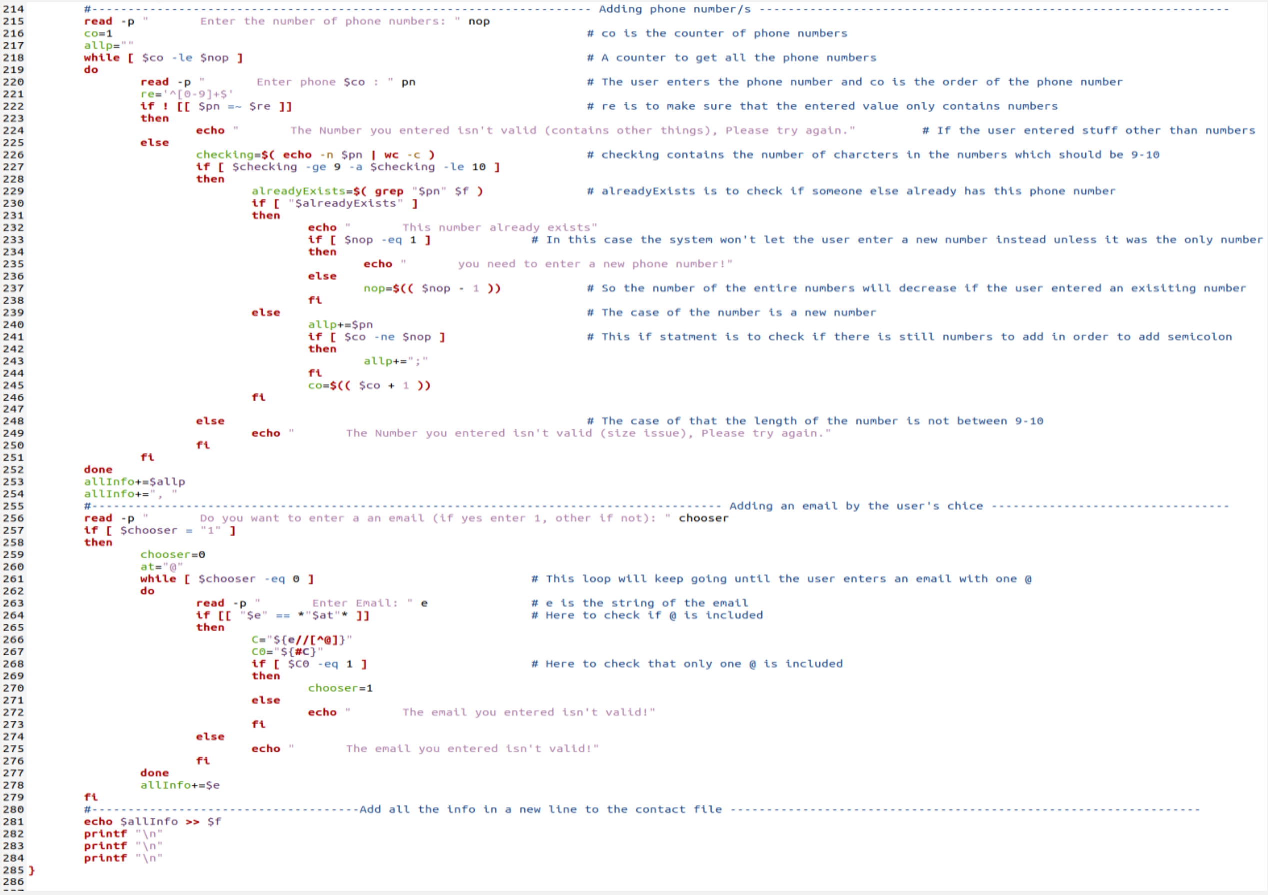
```
#==================================================== Main menu that first appears on the user screen ================================

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

if [ -e $f ]           # Only do the following if the file exists
then
        select=0                # The user menu at first
        printf "**** Welcome to Contact Managment System ****\n\n\n"
        printf "                Main Menu\n"
        printf "        ======================\n"
        printf "        [1] Add a new Contact\n"
        printf "        [2] List all Contects\n"
        printf "        [3] Search for Contact\n"
        printf "        [4] Edit a Contact\n"
        printf "        [5] Delete a Contact\n"
        printf "        [0] Exit\n"
        read  -p  "        Enter the choice: " select          # Here the user enters his selection
        m=(wc -l "$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\n"
                printf "                Main Menu\n"
                printf "        ======================\n"
                printf "        [1] Add a new Contact\n"
                printf "        [2] List all Contects\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
```

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.



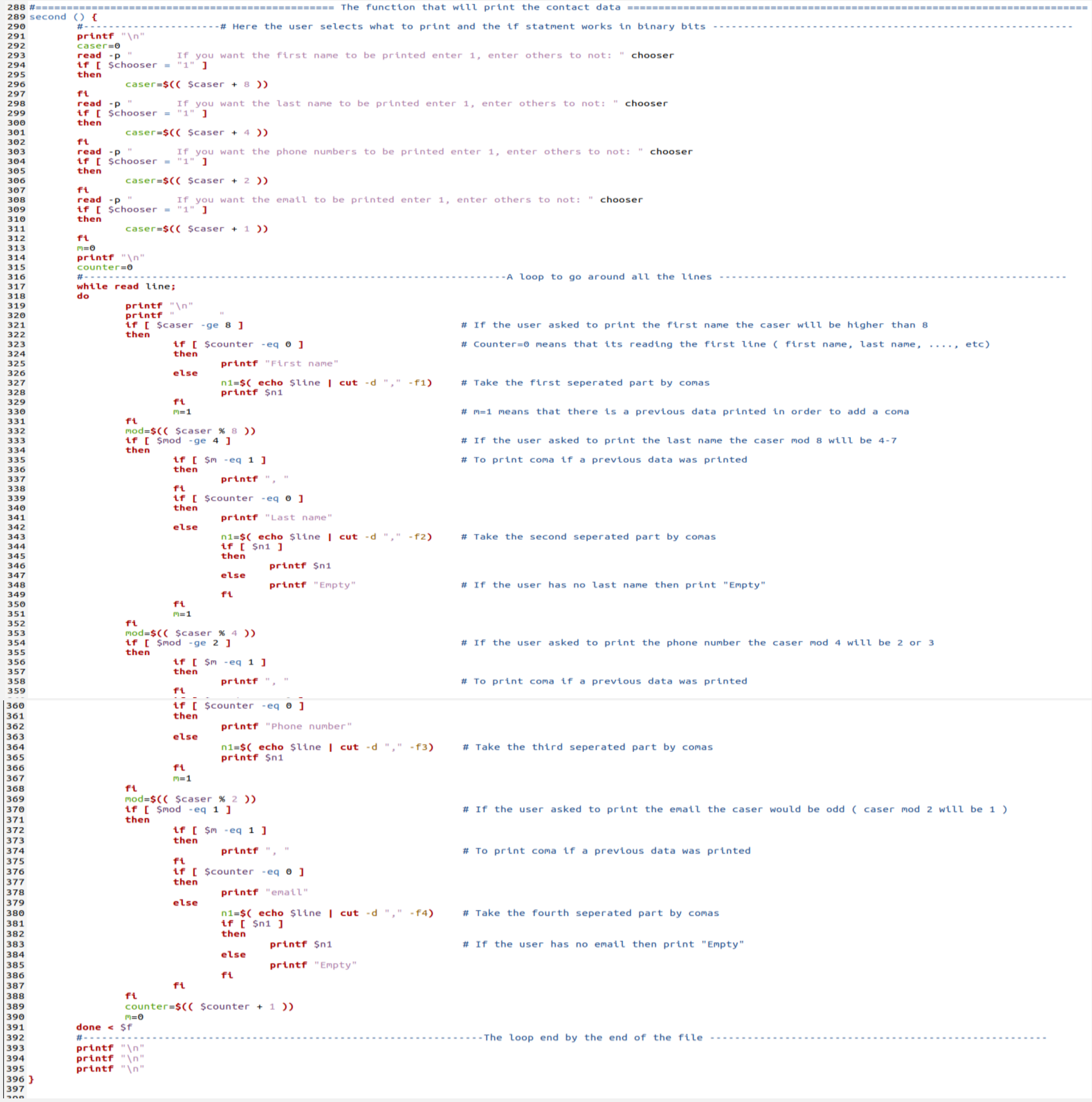
```
#================================================== The function that will add a new contact ==================================================================
first () {
        printf "\n"
        #-------------------------------------------------------------------------------------- Adding First 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 seperate 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"
}
```

2.3. The Second Function: Display All Contacts

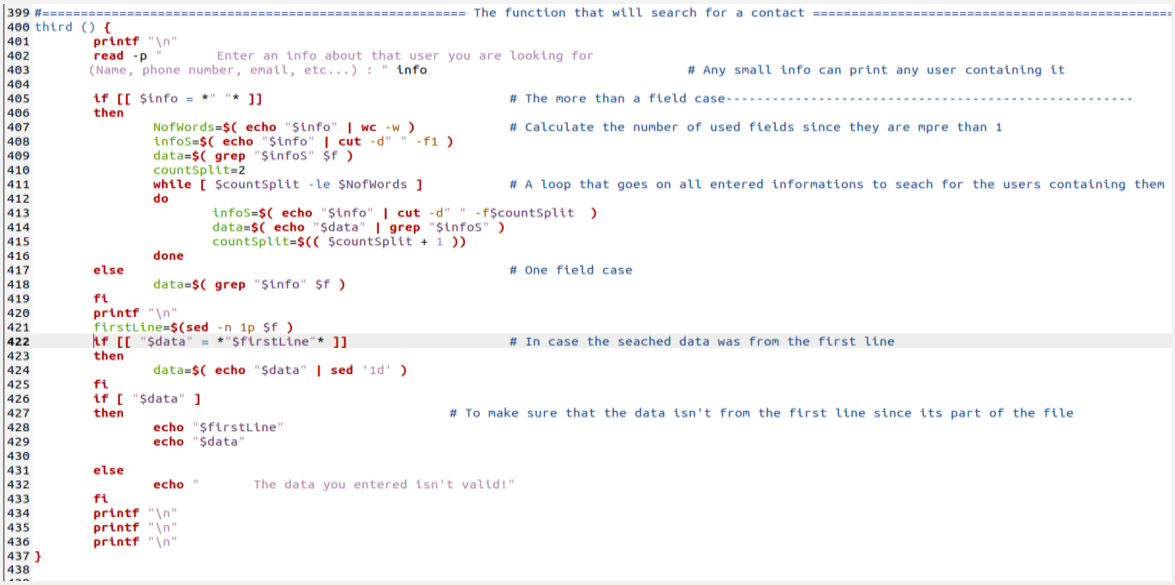
This function in to display all contacts where the user chooses if the field will be displayed or not, if the field is empty the system will print “Empty” and the user can choose to not print any field.



```
#============================================== The function that will print the contact data ======================================================================
second () {
        #---------------------# Here the user selects what to print and the if statment 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 mod 8 will be 4-7
                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 $n1
                                else
                                        printf "Empty"                  # If the user has no last name then print "Empty"
                                fi
                        fi
                        m=1
                fi
                mod=$(( $caser % 4 ))
                if [ $mod -ge 2 ]                                       # If the user asked to print the phone number the caser mod 4 will be 2 or 3
                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 $n1
                        fi
                        m=1
                fi
                mod=$(( $caser % 2 ))
                if [ $mod -eq 1 ]                                       # If the user asked to print the email the caser would be odd ( caser mod 2 will be 1 )
                then
                        if [ $m -eq 1 ]
                        then
                                printf ", "                             # To print coma if a previous data was printed
                        fi
                        if [ $counter -eq 0 ]
                        then
                                printf "email"
                        else
                                n1=$( 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"
}
```

2.4. The Third Function: Search for a Contact

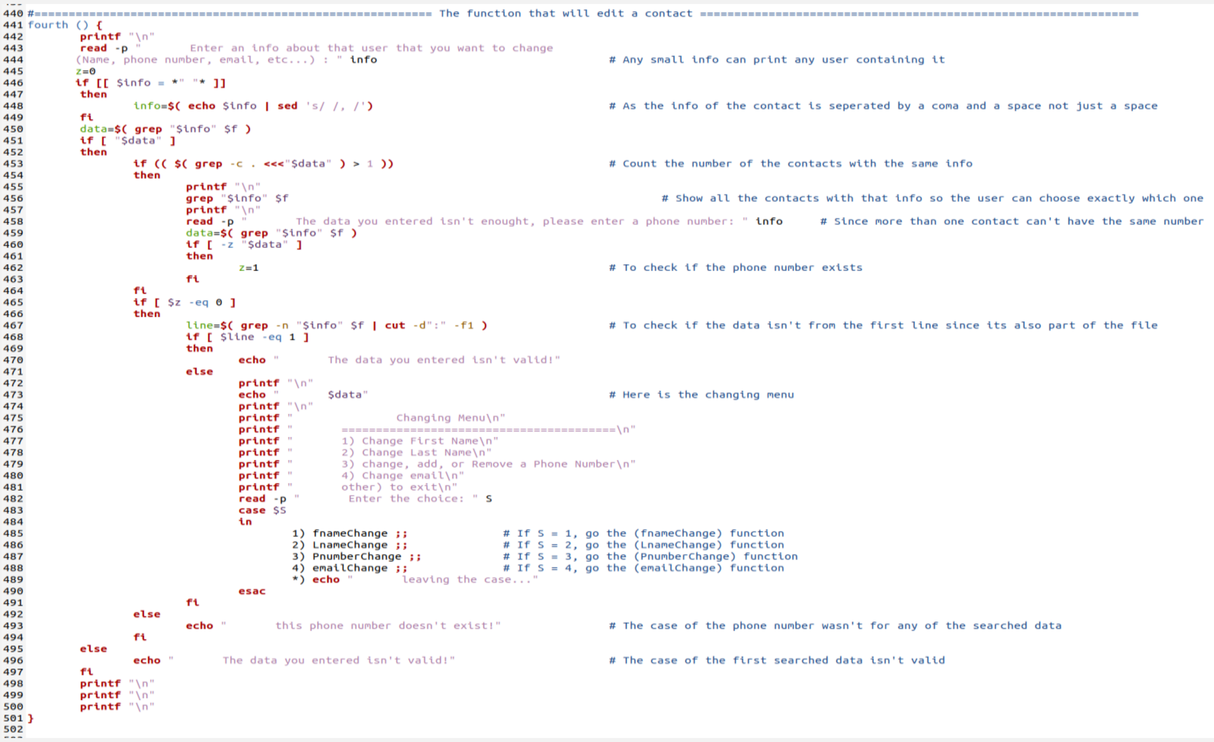
This function asks the user to enter the info that will be used to search for a user or users who contain the same info depending on one or more fields, if the info data doesn't match any Contact the system will tell the user that the data is invalid and it also makes sure that it's not for the first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



```
#======================================================= 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                             # Any small info can print any user containing it

        if [[ $info = *" "* ]]                                 # The more than a field case---------------------------------------------------
        then
                NofWords=$( echo "$info" | wc -w )             # Calculate the number of used fields since they are mpre than 1
                infoS=$( echo "$info" | cut -d" " -f1 )
                data=$( grep "$infoS" $f )
                countSplit=2
                while [ $countSplit -le $NofWords ]            # A loop that goes on all entered informations to seach for the users containing them
                do
                        infoS=$( echo "$info" | cut -d" " -f$countSplit  )
                        data=$( echo "$data" | grep "$infoS" )
                        countSplit=$(( $countSplit + 1 ))
                done
        else                                                   # One field case
                data=$( grep "$info" $f )
        fi
        printf "\n"
        firstLine=$(sed -n 1p $f )
        if [[ "$data" = *"$firstLine"* ]]                      # In case the seached data was from the first line
        then
                data=$( 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"
}
```

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.

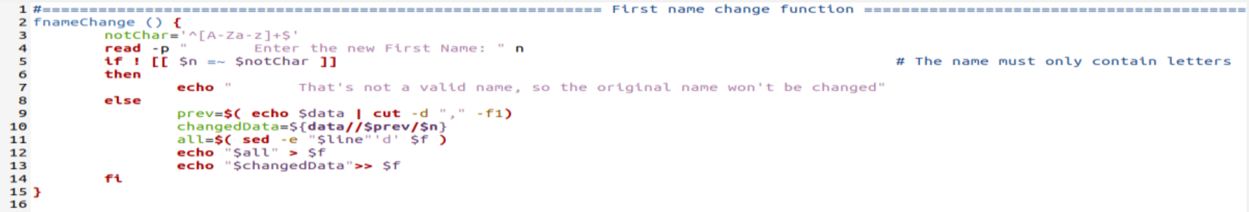


```
#========================================================== The function that will edit a contact =================================================================
fourth () {
        printf "\n"
        read -p "       Enter an info about that user that you want to change
       (Name, phone number, email, etc...) : " info                                     # Any small info can print any user containing it
       z=0
       if [[ $info = *" "* ]]
        then
                info=$( echo $info | sed 's/ /, /')                                     # As the info of the contact is seperated by a coma and a space not just a space
        fi
        data=$( 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" $f                                                         # Show all the contacts with that info so the user can choose exactly which one
                        printf "\n"
                        read -p "       The data you entered isn't enought, please enter a phone number: " info      # Since more than one contact can't have the same number
                        data=$( 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 ;;               # If S = 1, go the (fnameChange) function
                                        2) LnameChange ;;               # If S = 2, go the (LnameChange) function
                                        3) PnumberChange ;;             # If S = 3, go the (PnumberChange) function
                                        4) emailChange ;;               # If S = 4, go the (emailChange) function
                                        *) echo "       leaving the case..."
                                esac
                        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"
}
```

2.5.1.

2.5.1. The First Name Edit Function

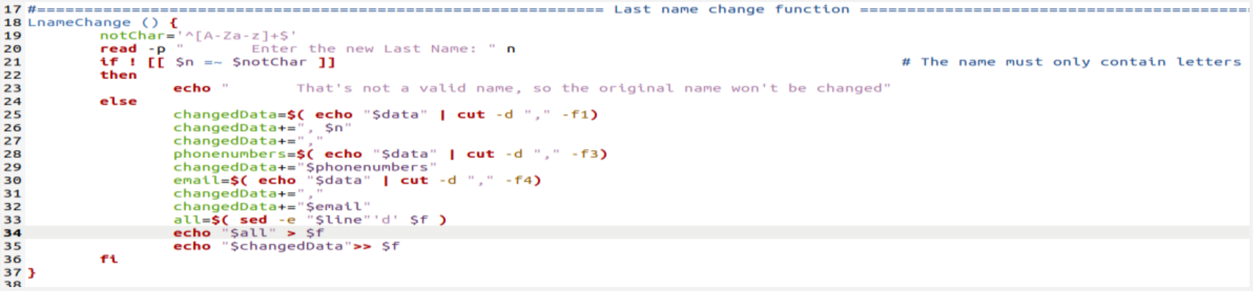
This function edits the first name or changes it and checks if the new name only contains alphabetic letters, if not it wont be edited.



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

2.5.2. The Last Name Edit Function

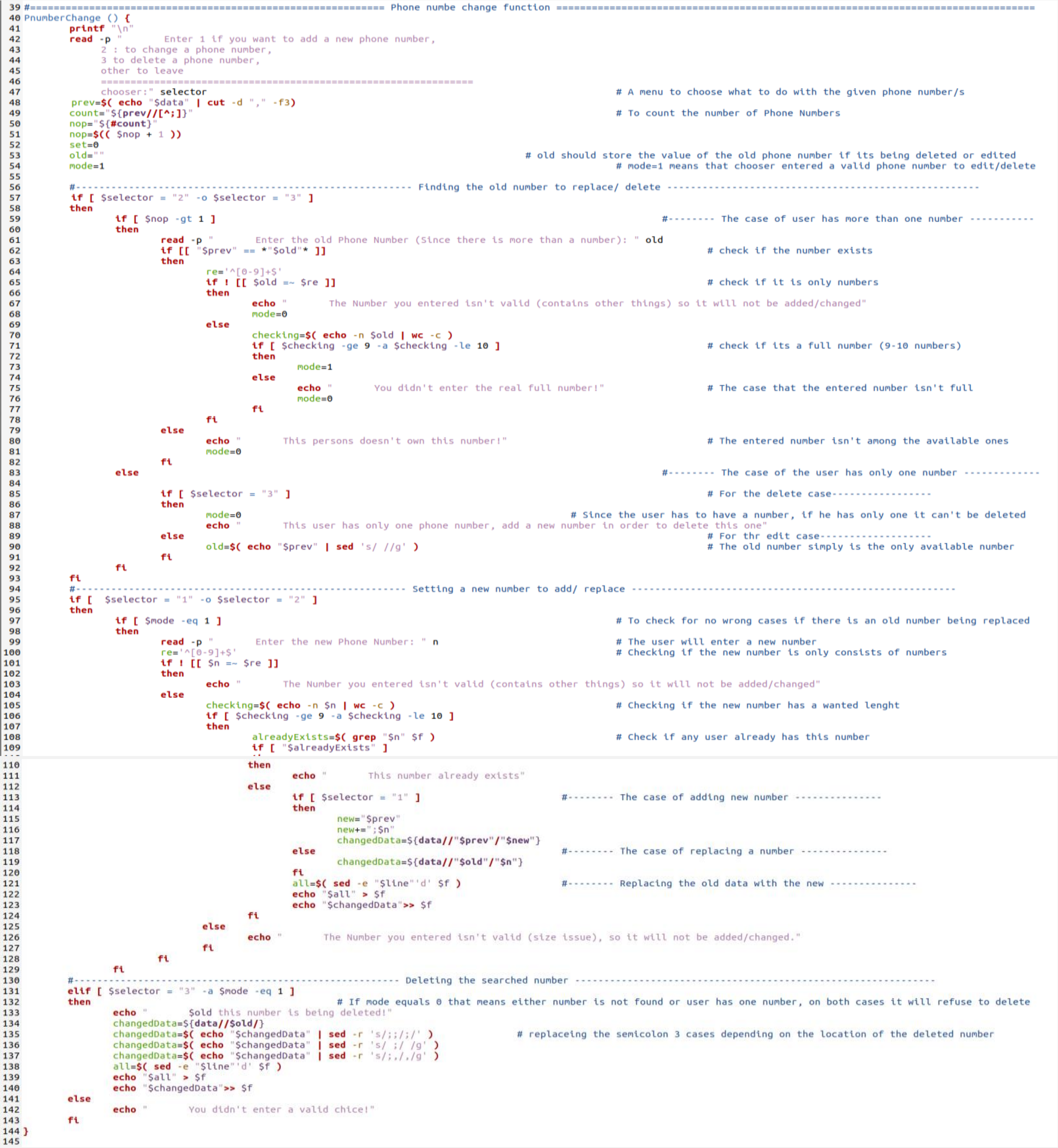
This function edits the second name and adds one if there isn’t and checks if it only contains alphabetic letters if not it won’t be edited.



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

2.5.3. The Phone Number Edit Function

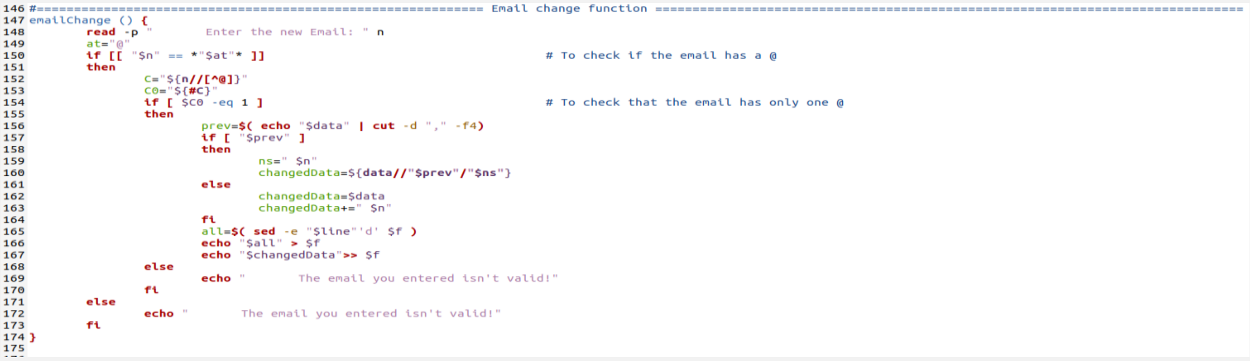
This function gives the user three options for the searched contact (add a new number, edit a number, delete a number) if the user is entering a new number or editing one then the system checks if the new number only contains numbers and that it is from 9-10 digits and if the user is editing or deleting 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.



```
#============================================================= 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                                                                 # A menu to choose what to do with the given phone number/s
         prev=$( echo "$data" | cut -d "," -f3)
        count="${prev//[^;]}"                                                                   # To count the number of Phone Numbers
        nop="${#count}"
        nop=$(( $nop + 1 ))
        set=0
        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 [ $selector = "3" ]                                                                # For the delete case----------------
                        then
                                mode=0                                                          # Since the user has to have a number, if he has only one it can't be deleted
                                echo "       This user has only one phone number, add a new number in order to delete this one"
                        else                                                                                  # For thr edit case------------------
                                old=$( echo "$prev" | sed 's/ //g' )                                          # The old number simply is the only available number
                        fi
                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 $checking -le 10 ]
                                then
                                        alreadyExists=$( grep "$n" $f )                         # Check if any user already has this number
                                        if [ "$alreadyExists" ]
                                        then
                                                echo "       This number already exists"
                                        else
                                                if [ $selector = "1" ]                          #-------- The case of adding new number --------------
                                                then
                                                        new="$prev"
                                                        new+=";$n"
                                                        changedData=${data//"$prev"/"$new"}
                                                else                                            #-------- The case of replacing a number --------------
                                                        changedData=${data//"$old"/"$n"}
                                                fi
                                                all=$( sed -e "$line"'d' $f )                   #-------- Replacing the old data with the new --------------
                                                echo "$all" > $f
                                                echo "$changedData">> $f
                                        fi
                                else
                                        echo "       The Number you entered isn't valid (size issue), so it will not be added/changed."
                                fi
                        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 "$changedData" | sed -r 's/;;/;/' )             # replaceing the semicolon 3 cases depending on the location of the deleted number
                changedData=$( 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
}
```

2.5.4. The Email Change Function

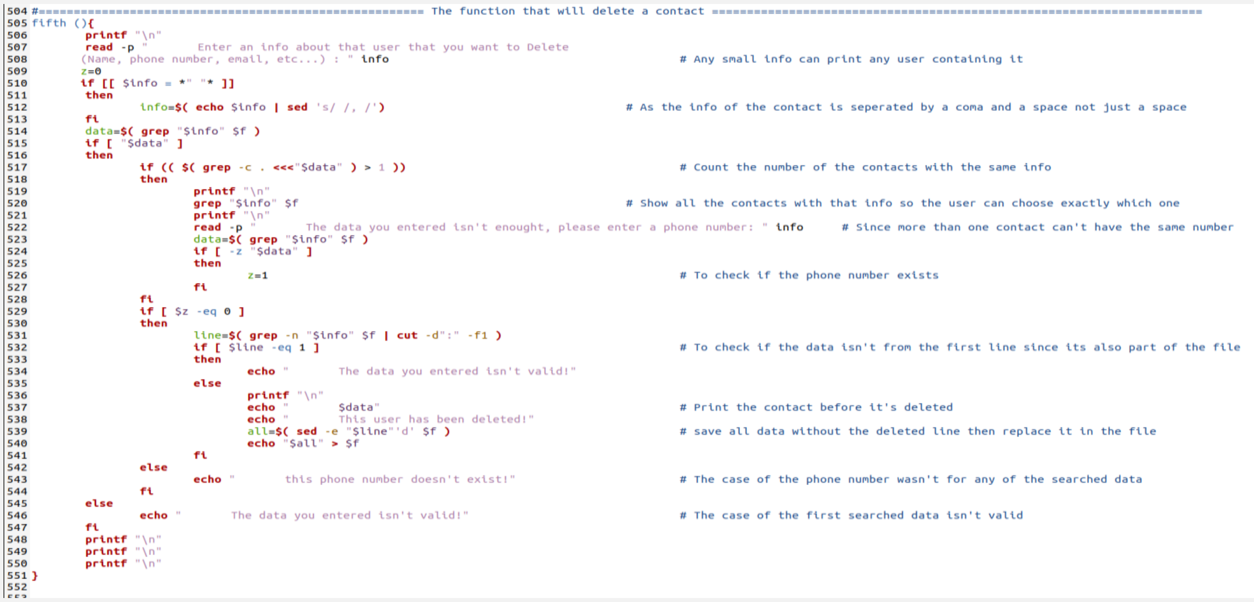
This function changes the email or adds a one if the contact didn’t previously have one and checks that it contains only one @.



```
#============================================================ Email change function ================================================================================
emailChange () {
        read -p "        Enter the new Email: " n
        at="@"
        if [[ "$n" == *"$at"* ]]                                              # To check if the email has a @
        then
                C="${n//[^@]}"
                C0="${#C}"
                if [ $C0 -eq 1 ]                                              # To check that the email has only one @
                then
                        prev=$( echo "$data" | cut -d "," -f4)
                        if [ "$prev" ]
                        then
                                ns=" $n"
                                changedData=${data//"$prev"/"$ns"}
                        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
}
```

2.6. The Fifth Function: Deleting a contact

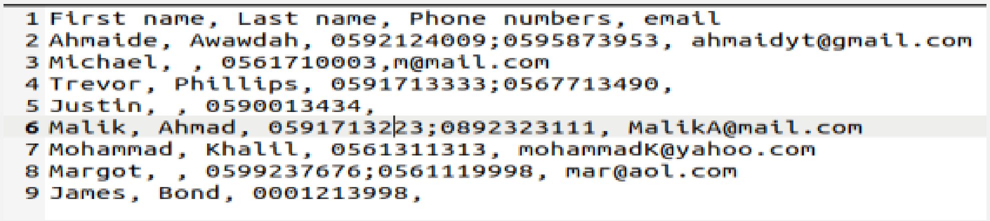
This function searches for a contact based on given info, if the info is already there then the system will ask the user to enter the phone number and then it will print the contact before deleting it, then it deletes it.



```
#======================================================= 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" $f                                       # Show all the contacts with that info so the user can choose exactly which one
                        printf "\n"
                        read -p "        The data you entered isn't enought, please enter a phone number: " info      # Since more than one contact can't have the same number
                        data=$( 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 )
                        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
                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"
}
```

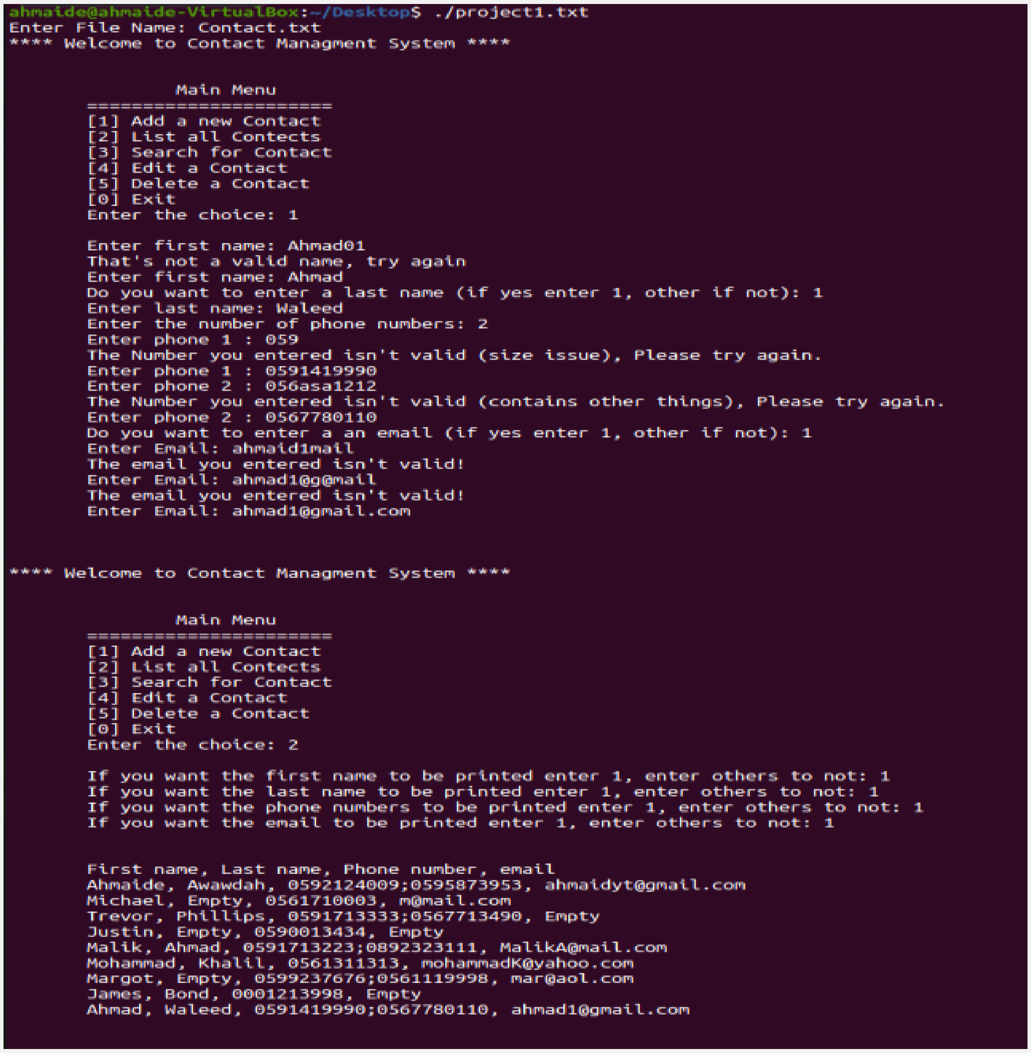
3. Testing Examples

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

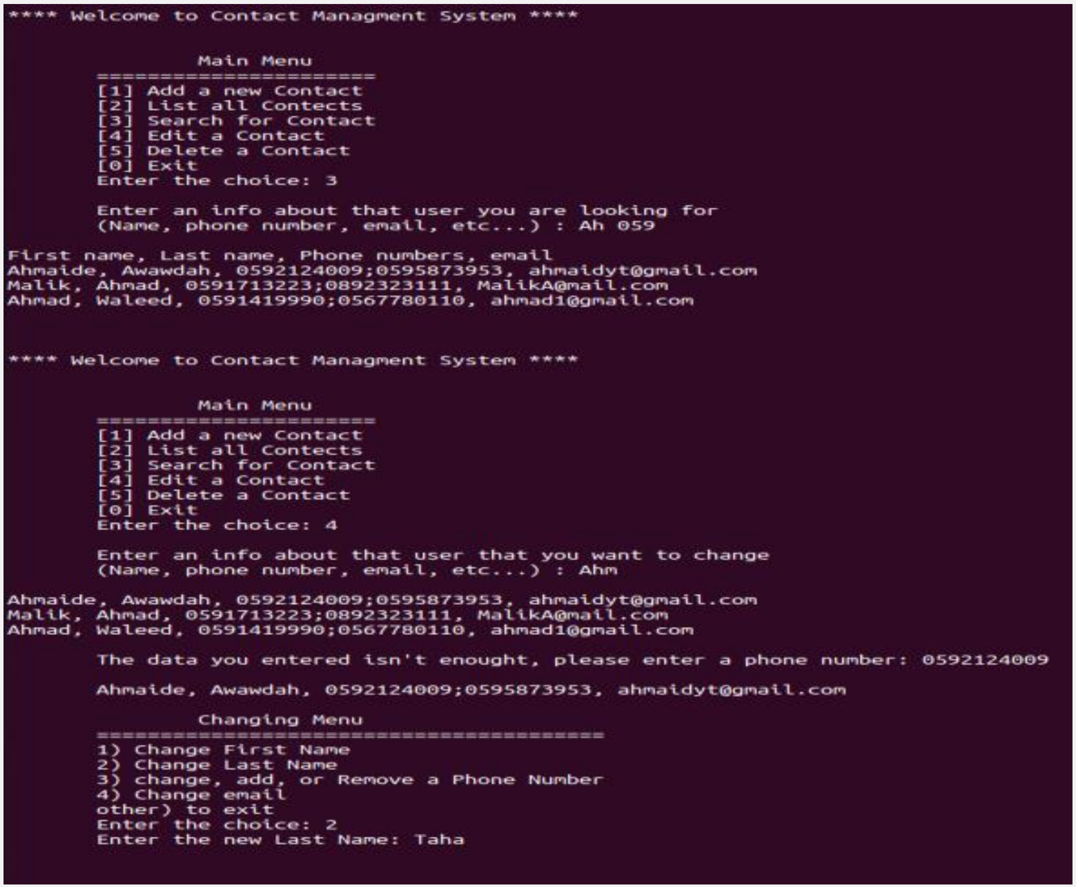


```
First name, Last name, Phone numbers, email
Ahmaide, Awawdah, 0592124009;0595873953, ahmaidyt@gmail.com
Michael, , 0561710003,m@mail.com
Trevor, Phillips, 0591713333;0567713490,
Justin, , 0590013434,
Malik, Ahmad, 0591713223;0892323111, MalikA@mail.com
Mohammad, Khalil, 0561311313, mohammadK@yahoo.com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

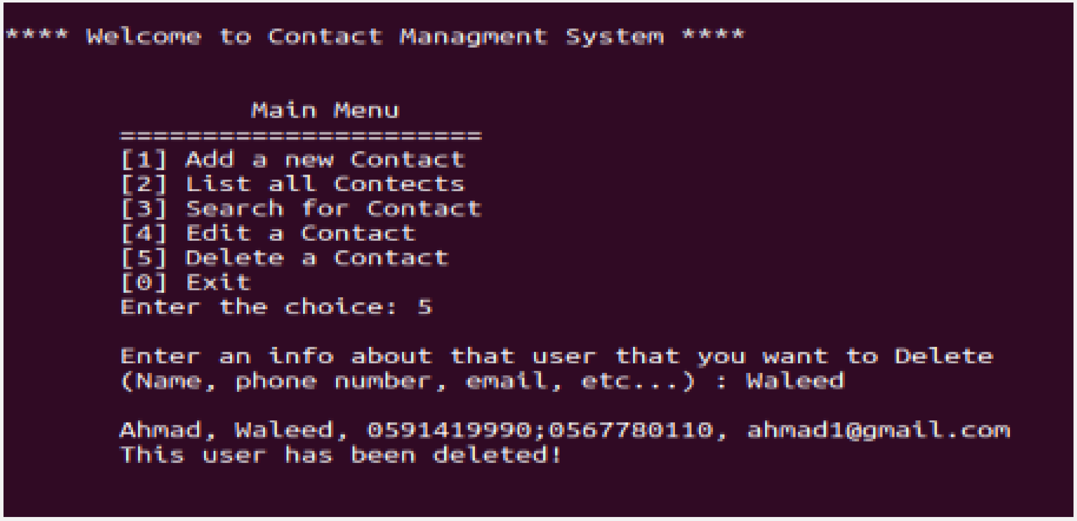
3.1. Test Example 1



```
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 Contects
        [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 an email (if yes enter 1, other if not): 1
        Enter Email: ahmaid1mail
        The email you entered isn't valid!
        Enter Email: ahmad1@g@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 Contects
        [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, ahmaidyt@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
```

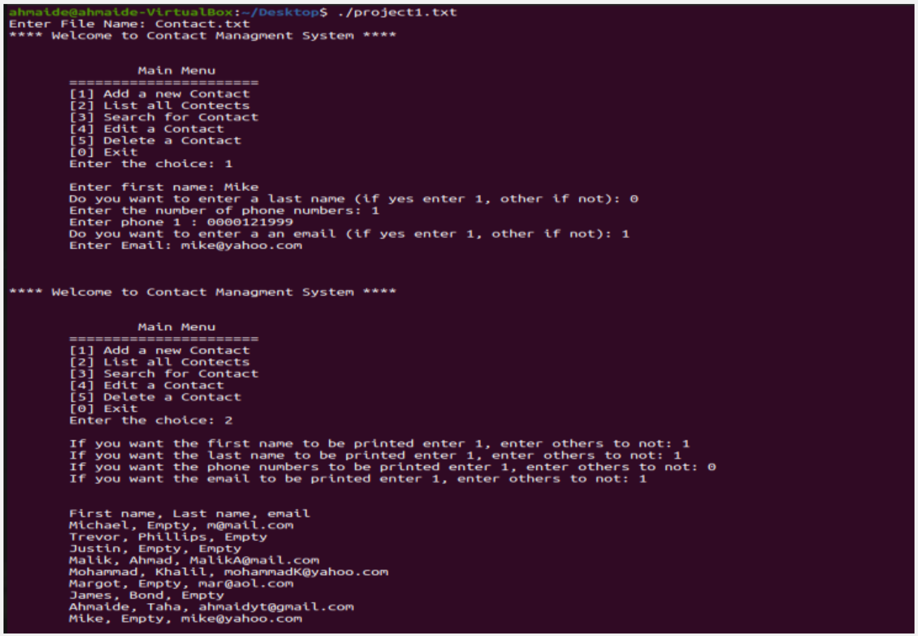


```
**** Welcome to Contact Managment System ****

               Main Menu
        ======================
        [1] Add a new Contact
        [2] List all Contects
        [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, ahmaidyt@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 Contects
        [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, ahmaidyt@gmail.com
Malik, Ahmad, 0591713223;0892323111, MalikA@mail.com
Ahmad, Waleed, 0591419990;0567780110, ahmad1@gmail.com

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

        Ahmaide, Awawdah, 0592124009;0595873953, ahmaidyt@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
```

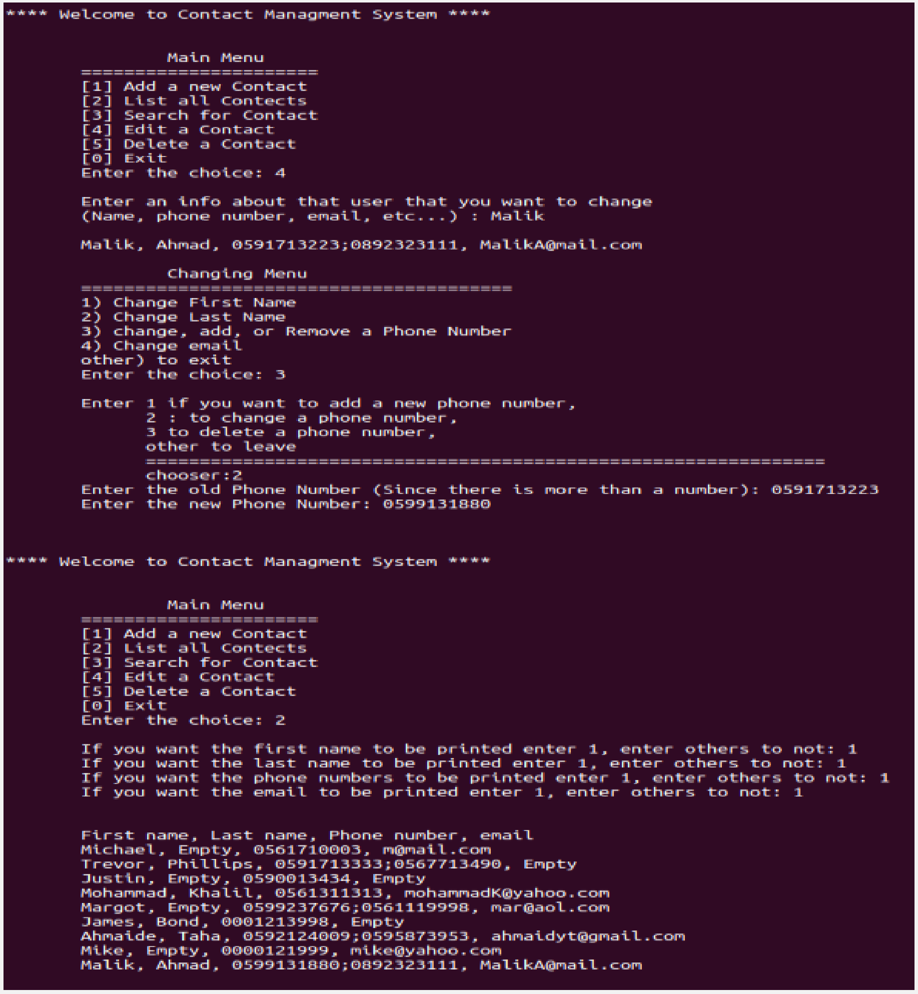


```
**** Welcome to Contact Managment System ****

               Main Menu
        ======================
        [1] Add a new Contact
        [2] List all Contects
        [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!
```

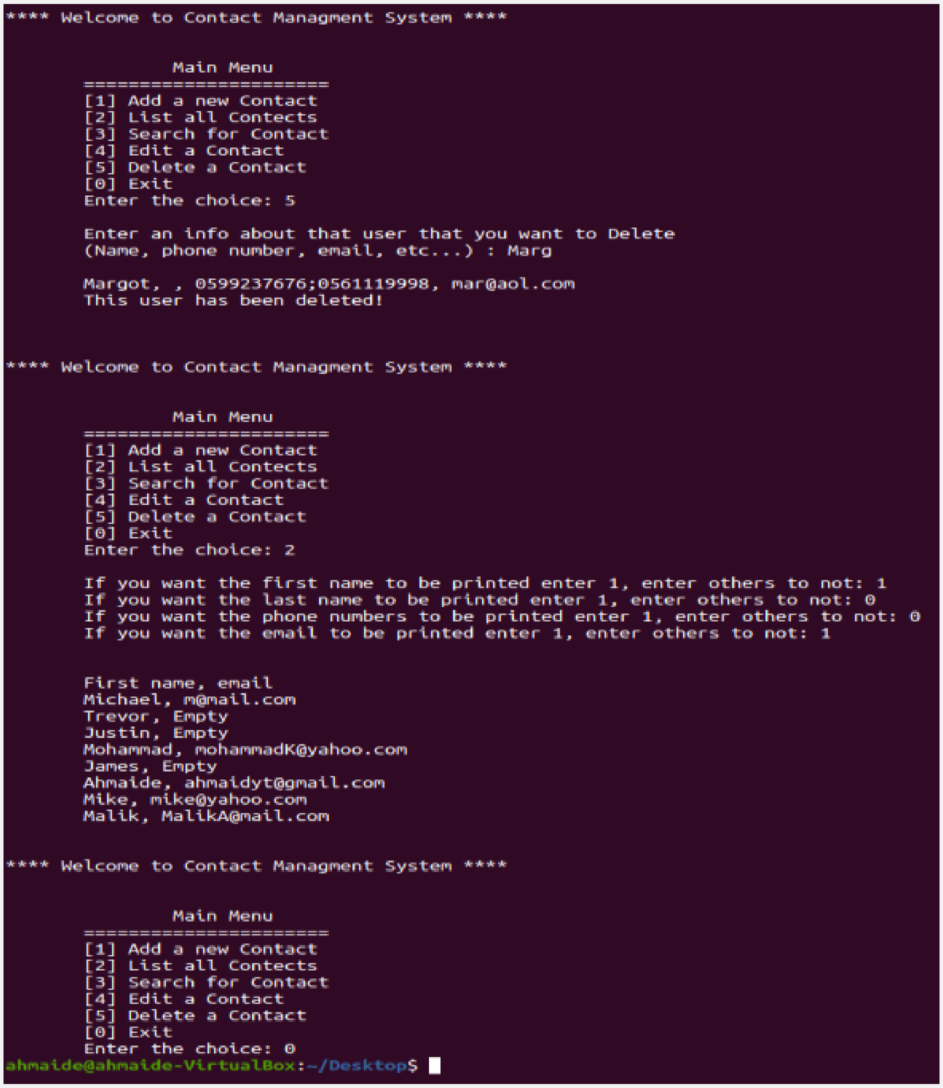
3.2. Test Example 2



```
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 Contects
        [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 : 0000121999
        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 Contects
        [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, ahmaidyt@gmail.com
        Mike, Empty, mike@yahoo.com
```



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


               Main Menu
        ======================
        [1] Add a new Contact
        [2] List all Contects
        [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 Contects
        [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, ahmaidyt@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 Contects
        [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 Contects
        [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, ahmaidyt@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 Contects
        [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 =~ $notChar ]]                                                # The name must only contain letters
        then
                echo "      That's not a valid name, so the original name won't be changed"
        else
                prev=$( 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 =~ $notChar ]]                                                # The name must only contain letters
        then
                echo "      That's not a valid name, so the original name won't be changed"
        else
                changedData=$( echo "$data" | cut -d "," -f1)
                changedData+=", $n"
                changedData+=","
                phonenumbers=$( echo "$data" | cut -d "," -f3)
                changedData+="$phonenumbers"
                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                                          # A menu to choose what to do with the given phone number/s
   prev=$( echo "$data" | cut -d "," -f3)
        count="${prev//[^;]}"                                                    # To count the number of Phone Numbers
        nop="${#count}"
        nop=$(( $nop + 1 ))
        set=0
```

```
    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 [ $selector = "3" ]                          # For the delete case-----------------
            then
                mode=0
                # Since the user has to have a number, if he has only one it can't be deleted
                echo "     This user has only one phone number, add a new number in order to delete this one"
            else                                                        # For thr edit case-------------------
                old=$( echo "$prev" | sed 's/ //g' )        # The old number simply is the only available number
            fi
        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 $checking -le 10 ]
                then
                    alreadyExists=$( grep "$n" $f )                         # Check if any user
already has this number
                    if [ "$alreadyExists" ]
                    then
                        echo "      This number already exists"
                    else
                        if [ $selector = "1" ]                              #-------- The case of
adding new number ---------------
                        then
                            new="$prev"
                            new+=";$n"
                            changedData=${data//"$prev"/"$new"}
                        else                                                    #--------
The case of replacing a number ---------------
                            changedData=${data//"$old"/"$n"}
                        fi
                        all=$( sed -e "$line"'d' $f )                       #-------- Replacing the
old data with the new ---------------
                        echo "$all" > $f
                        echo "$changedData">> $f
                    fi
                else
                    echo "      The Number you entered isn't valid (size issue), so it will not be
added/changed."
                fi
            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 "$changedData" | sed -r 's/;;/;/' )         # replaceing the semicolon 3 cases
depending on the location of the deleted number
        changedData=$( 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="${n//[^@]}"
        C0="${#C}"
        if [ $C0 -eq 1 ]                        # To check that the email has only one @
        then
            prev=$( echo "$data" | cut -d "," -f4)
            if [ "$prev" ]
            then
                ns=" $n"
```

```
                                changedData=${data//"$prev"/"$ns"}
                        else
                                changedData=$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
}


#=================================================== 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 seperate 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 statment 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
mod 8 will be 4-7
        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 $n1
                else
                    printf "Empty"                  # If the user has no last name then print
"Empty"
                fi
            fi
            m=1
        fi
        mod=$(( $caser % 4 ))
        if [ $mod -ge 2 ]                       # If the user asked to print the phone number the
caser mod 4 will be 2 or 3
        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 $n1
            fi
            m=1
        fi
        mod=$(( $caser % 2 ))
        if [ $mod -eq 1 ]                       # If the user asked to print the email the caser would
be odd ( caser mod 2 will be 1 )
        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
```

```
                                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                                        # Any small info can print any user
containing it
        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 $info | sed 's/ /, /')                                    # As the info of the contact is seperated by
a coma and a space not just a space
        fi
        data=$( 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" $f                                                        # Show all the
contacts with that info so the user can choose exactly which one
                        printf "\n"
                        read -p "       The data you entered isn't enought, please enter a phone number: " info        # Since more than
one contact can't have the same number
                        data=$( 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 ;;               # If S = 1, go the (fnameChange) function
                                        2) LnameChange ;;               # If S = 2, go the (LnameChange) function
                                        3) PnumberChange ;;             # If S = 3, go the (PnumberChange) function
                                        4) emailChange ;;               # If S = 4, go the (emailChange) function
                                        *) echo "       leaving the case..."
                                esac
                        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" $f                                                  # Show all the contacts with that
info so the user can choose exactly which one
                        printf "\n"
                        read -p "      The data you entered isn't enought, please enter a phone number: " info      # Since more than
one contact can't have the same number
                        data=$( 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 )
                        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
                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 $f ]            # Only do the following if the file exists
then
        select=0                # The user menu at first
        printf "**** Welcome to Contact Managment System ****\n\n\n"
        printf "            Main Menu\n"
        printf "      ======================\n"
        printf "      [1] Add a new Contact\n"
        printf "      [2] List all Contects\n"
        printf "      [3] Search for Contact\n"
        printf "      [4] Edit a Contact\n"
        printf "      [5] Delete a Contact\n"
        printf "      [0] Exit\n"
        read  -p  "      Enter the choice: " select                 # Here the user enters his selection
        m=(wc -l "$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\n"
                printf "            Main Menu\n"
                printf "      ======================\n"
                printf "      [1] Add a new Contact\n"
                printf "      [2] List all Contects\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
```