



BIRZEIT UNIVERSITY

Faculty of Engineering and Technology
Electrical and Computer Engineering Department
ENCS 3130
Linux Laboratory
Shell Scripting Project Report

Instructor's Name: Dr. Khader Mohammad.

Student's Name-ID:

Lojain Abdalrazaq-1190707

Aseel Deek-1190587

Section: 2.

Date: Sunday, 1 Dec 2021.

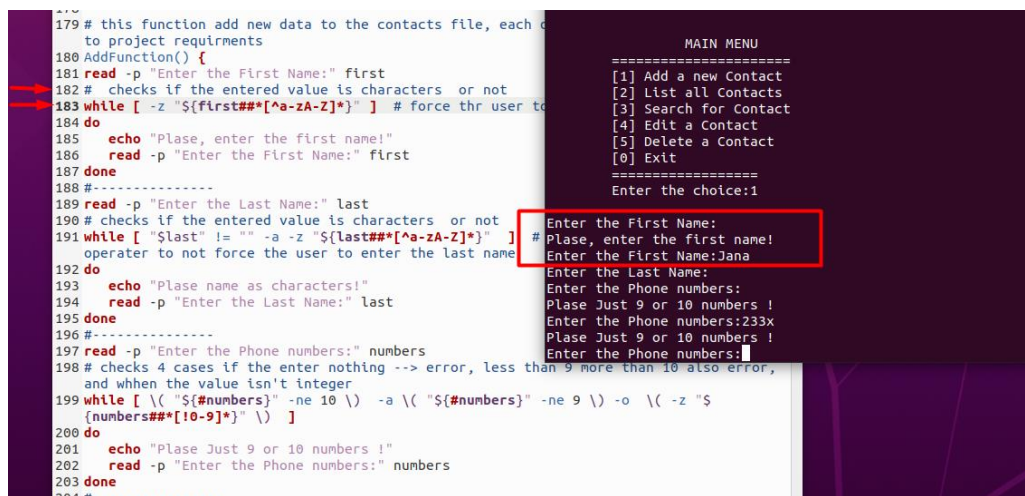
Introduction:

This project was created using shell scripting to manipulate a file text called **Contact.txt**. Some operation could be used to manipulate the file like (adding new data to the file, change the current data and the new added one, delete the unwanted data, list them and search for any filed contains in the file text. All these operations written with specific rules that mange the file in a logical way.

▪ Adding:

For example, in adding operation, the user **Must** enter the first name and the phone number at least. Figure (1.1) shows this test case with the code.

also, the program validates the input data. Figure (1.2) shows the output when entering value in the number section is not integers.

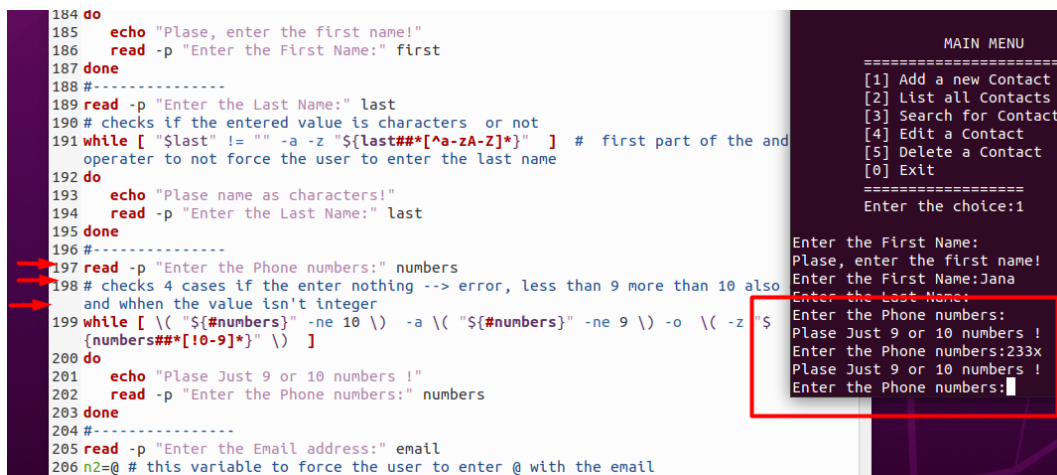


```
179 # this function add new data to the contacts file, each d
to project requirments
180 AddFunction() {
181 read -p "Enter the First Name:" first
182 # checks if the entered value is characters or not
183 while [ -z "${first##*[!a-zA-Z]*}" ] # force thr user to
184 do
185     echo "Plase, enter the first name!"
186     read -p "Enter the First Name:" first
187 done
188 #-----
189 read -p "Enter the Last Name:" last
190 # checks if the entered value is characters or not
191 while [ "$last" != "" -a -z "${last##*[!a-zA-Z]*}" ] #
operator to not force the user to enter the last name
192 do
193     echo "Plase name as characters!"
194     read -p "Enter the Last Name:" last
195 done
196 #-----
197 read -p "Enter the Phone numbers:" numbers
198 # checks 4 cases if the enter nothing --> error, less than 9 more than 10 also error,
and when the value isn't integer
199 while [ \("${numbers}" -ne 10 \) -a \("${numbers}" -ne 9 \) -o \(-z "${
numbers##*[!0-9]*}" \) ]
200 do
201     echo "Plase Just 9 or 10 numbers !"
202     read -p "Enter the Phone numbers:" numbers
203 done
204 #-----
```

```
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 the First Name:
Plase, enter the first name!
Enter the First Name:Jana
Enter the Last Name:
Enter the Phone numbers:
Plase Just 9 or 10 numbers !
Enter the Phone numbers:233x
Plase Just 9 or 10 numbers !
Enter the Phone numbers:
```

Figure (1.1): shows the first test case with its code



```
184 do
185     echo "Plase, enter the first name!"
186     read -p "Enter the First Name:" first
187 done
188 #-----
189 read -p "Enter the Last Name:" last
190 # checks if the entered value is characters or not
191 while [ "$last" != "" -a -z "${last##*[!a-zA-Z]*}" ] # first part of the and
operator to not force the user to enter the last name
192 do
193     echo "Plase name as characters!"
194     read -p "Enter the Last Name:" last
195 done
196 #-----
197 read -p "Enter the Phone numbers:" numbers
198 # checks 4 cases if the enter nothing --> error, less than 9 more than 10 also
and when the value isn't integer
199 while [ \("${numbers}" -ne 10 \) -a \("${numbers}" -ne 9 \) -o \(-z "${
numbers##*[!0-9]*}" \) ]
200 do
201     echo "Plase Just 9 or 10 numbers !"
202     read -p "Enter the Phone numbers:" numbers
203 done
204 #-----
205 read -p "Enter the Email address:" email
206 n2=@ # this variable to force the user to enter @ with the email
---
```

```
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 the First Name:
Plase, enter the first name!
Enter the First Name:Jana
Enter the Last Name:
Enter the Phone numbers:
Plase Just 9 or 10 numbers !
Enter the Phone numbers:233x
Plase Just 9 or 10 numbers !
Enter the Phone numbers:
```

Figure (1.2): shows the second test case with its code

▪ Searching:

In searching function, we have declared a test case that checks the input data for example if the entered data was “aseel deek” → change it to “Aseel, Deek”. Figure 1.3 shows an example.

```

24 fi # end of the 1st if
25
26 done # end of while loop
27
28 }
29 #----- Search Function -----
30
31 # this function asks for string to search for
32 SearchFunction() {
33   read -p "Enter the string you are looking for: " str
34   at=@ #variable
35   str=$(echo "$str" | sed 's/ /, /g' ) # this case for entering "aseel deek"
36   repalce it by ,space
37   if ( echo "$str" | grep -q "$at" ) # this case to prvent changing the mail
38   then
39     if grep -q "$str" "$file"; then
40       echo "$str was found\n"
41     else
42       echo "$str was not found1\n"
43     fi
44   elif ! ( echo "$str" | grep -q "$at" ) # this case is the oposite
45   then
46     str=$( echo "$str" | sed -r 's/\<./\U&/g' ) # convert the firsr letter in each word to upper
47     if grep -q "$str" "$file"; then # if these words exist, then print the found message
48       echo "$str was found\n"
49     else
50       echo "$str was not found3\n"
51     fi

```

```

[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
=====
Enter the choice:3
Enter the string you are looking for: aseel deek
Aseel, Deek was found

**** 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 (1.3): shows the case when searching for a string

▪ Deleting:

In this function, the file contain will be shown, and the user just enter the line number to delete the specified data.

```

1 First name, Last Name, Phone numbers, email
2 Aziz, Qaroush, 0599357111;092991111, aqaroush@birzeit.edu
3 Aseel, Deek, 05994875, aswe@
4 Ahmad, Khalil, 0569345673, khalil@gmail.com
5 Zaid, Khamis, 0254123612, zaidK@gmail.com
6 Lojain, jalal, 0000024512, lojain@gmail.com
7 Ward, Odeh, 1111112212. ward@birzeit.edu

```

```

[0] Exit
=====
Enter the choice:5
1 First name, Last Name, Phone numbers, email
2 Aziz, Qaroush, 0599357111;092991111, aqaroush@birzeit.edu
3 Aseel, Deek, 05994875, aswe@
4 Ahmad, Khalil, 0569345673, khalil@gmail.com
5 Zaid, Khamis, 0254123612, zaidK@gmail.com
6 Jana, , 123456987, dsdf@
7 Lojain, jalal, 0000024512, lojain@gmail.com
8 Ward, Odeh, 1111112212. ward@birzeit.edu

Enter the Line Numfber you want to Remove: 6
Line 6 has been deleted

**** Welcome to Contact Management System ****

```

Figure (1.4): shows when deleting a line in the contact file

▪ Editing:

This function displays the content of the Contacts.txt file for the user with the line number and the user must choose the line number that he want to make edit on it, and when the user enter **invalid** number of lines, the program asks him to enter a valid line number to make changes on it as shown in the following figure **fig(1)**. After that the edited data will be saved in the Contacts.txt file.

```

#-----> EditFunction <-----
EditFunction(){
    echo "\n"
    echo "*****"
    echo "\n"
    echo "\tHello, Here is the content of \"$file\":"
    echo "\n"
    cat -n "$file"
    echo "\n"
    echo "*****"

    #Calculate the number of contacts in the txt file
    num=$(cat -n "$file" | wc -l)

    #
    read -p "Please Enter the Number of the line to edit in it : " LineNumber
    #while loop will continue until the user enter a valid line number
    while [ "$LineNumber" -gt "$num" ]
    do
        #Error message when the user enter invlid line number to edit in it
        read -p "Invalid Number, Please Enter Again: " LineNumber
    done
    #Displaying the line that the user has chose
    sed -n "$LineNumber p" "$file"

    #Save the new data in the Contacts.txt file
}

```

```

*****
Hello, Here is the content of Contacts.txt:

1 Lojain, Aseel, 123456789, lojain.jalal@gmail.com
2 Aziz, Qaroush, 123456789, aqaroush@birzeit.com
3 Ahmad, Khalil, 0569345673, khalil@gmail.com

*****
Please Enter the Number of the line to edit in it : 4
Invalid Number, Please Enter Again: 6
Invalid Number, Please Enter Again: 5
Invalid Number, Please Enter Again: 1
Lojain, Aseel, 123456789, lojain.jalal@gmail.com
Enter the value to edit: Lojain
[ Lojain ] is found....
Enter the edited (new) value: 34lojain
Enter the edited value only (chars): lojain.ja
Enter the edited value only (chars): Newlojain
The data is edited succesfully.
** Welcome to Contact Management System ***

```

Fig(1.5): shows the first test case with its code.

When the user want to change the name for example, the new value must be also name with characters **only**, and the same thing with the phone number and the email address, the new value of the phone number will be numbers only (**with length of 9 or 10**) and the email address must contain the **@ symbol**.

```

*****PHONE NUMBERS*****

#if the input is number
elif ! [ -z "${str##*[!0-9]*}" ]; then
    read -p "Enter the edited (new) value: " editNum
    #while loop will keep until the value of the edited number is valid --> only numbers
    while [ \{ "${editNum}" -ne 10 \} -a \{ "${editNum}" -ne 9 \} -o \{ -z "${editNum##*[!0-9]*}" \} ]
    do
        read -p "Enter the edited value (only number) & at least 9 numbers and 10 at most: " editNum
    done
    #get out the loop when its valid
    #saving the data in the file
    sed -e 's/\"$str\"/'$editNum'/g' "$file" < "$file" > temp1.txt
    cp "temp1.txt" "$file"
    echo "the data is edited"
#-----> EMAIL <-----
*****

```

```

Hello, Here is the content of Contacts.txt:

1 Newlojain, Aseel, 123456789, lojain.jalal@gmail.com
2 Aziz, Qaroush, 123456789, aqaroush@birzeit.com
3 Ahmad, Khalil, 0569345673, khalil@gmail.com

*****
Please Enter the Number of the line to edit in it : 1
Newlojain, Aseel, 123456789, lojain.jalal@gmail.com
Enter the value to edit: 123456789
[ 123456789 ] is found....
Enter the edited (new) value: 1432
Enter the edited value (only number) & at least 9 numbers and 10 at most: 2145
Enter the edited value (only number) & at least 9 numbers and 10 at most: 987654321
the data is edited
** Welcome to Contact Management System ***

```

Fig(1.6): shows the second test case with its code.

When the value that we want to edit is not in the line, there will be an error message to tell the user that the value is **not found** in the selected line as shown in the following figure

```

6 write : ( echo $editEmail | grep -q $del )
7 do
8 read -p "Enter the edited value (must have @ sign):" editEmail
9 done
10 #get out the loop when its valid
11 #saving the data in the temp.txt file
12 sed -e 's/'$str'/'$editEmail'/g' "$file" < "$file" > temp1.txt
13 cp "temp1.txt" "$file"
14 echo "the data is edited"
15
16 fi
17 *****NO MATCH*****
18 # -----NOT FOUND-----
19 # if the input data is not found in the file
20 else
21 #printing an error message that its not found
22 echo " [ $str ] is NOT found\n"
23 fi
24
25 }
26
27 # -----Delete Function -----
28

```

```

*****
Hello, Here is the content of Contacts.txt:

1 Newlojain, Aseel, 987654321, lojain.jalal@gmail.com
2 Aziz, Qaroush, 987654321, qaroush@birzeit.com
3 Ahmad, Khalil, 0569345673, khalil@gmail.com

*****
Please Enter the Number of the line to edit in it : 2
Aziz, Qaroush, 987654321, qaroush@birzeit.com
Enter the value to edit: 123456789
[ 123456789 ] is NOT found

** Welcome to Contact Management System ****

```

Fig(1.7): shows the third test case with its code.

▪ Listing:

In this function, the contacts will be listed according to first name, last name or both (first name and last name). The user determines which form he want.

```

ListFunction(){
#The user supposed to enter the choice in order to list the contacts
echo "Hi, Based ON what you want to list the Contacts Information? Press (F) fot the first name ,(L) for the last name / Both( B )?"
#L-->List Name, F-->First Name, B-->Both Cols(Last &First)

read ListChoice
#reading the input from the user
#If the input was non of the option
#the user will be asked to enter a valid choice for listing the contacts

while [ "$ListChoice" != "L" -a "$ListChoice" != "F" -a "$ListChoice" != "B" -a "$ListChoice" != "l" -a "$ListChoice" != "f" -a "$ListChoice" != "b" ]
do
#while loop will keep working until the user enter a valid option

read -p "Bad choice, Please Enter a Valid choice: [F/L/B]: " ListChoice
done

#-----LAST NAME-----
#Last name
# Listing based on the last Name
if [ "$ListChoice" = "L" -o "$ListChoice" = "l" ]
then
echo "Last"
echo "-----"
#the sed command for removing the comma between the elements
sed -e 's/,/t/g' "$file" | cut -d ' ' -f2

#-----FIRST NAME-----
#First Name
# Listing based on the flrst Name
elif [ "$ListChoice" = "F" -o "$ListChoice" = "f" ]
then
echo "First"
echo "-----"

```

```

[2] List all Contacts
[3] Search for Contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
*****
Enter the choice:2

Hi, Based ON what you want to list the Contacts Information? Press (F) fot the first name ,(L) for the last name / Both( B )?

First
-----
Newlojain
Aziz
Ahmad
** 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:2

Hi, Based ON what you want to list the Contacts Information? Press (F) fot the first name ,(L) for the last name / Both( B )?

Last
-----
Aseel
Qaroush
Khalil
** Welcome to Contact Management System ****

```

Fig(1.8): shows when listing contacts based on first or last name.