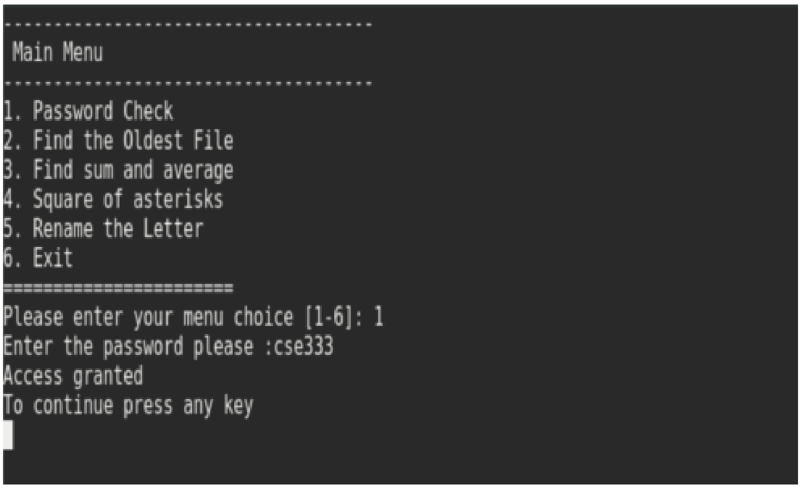
REPORT OF OPERATING SYSTEM FIRST PROJECT

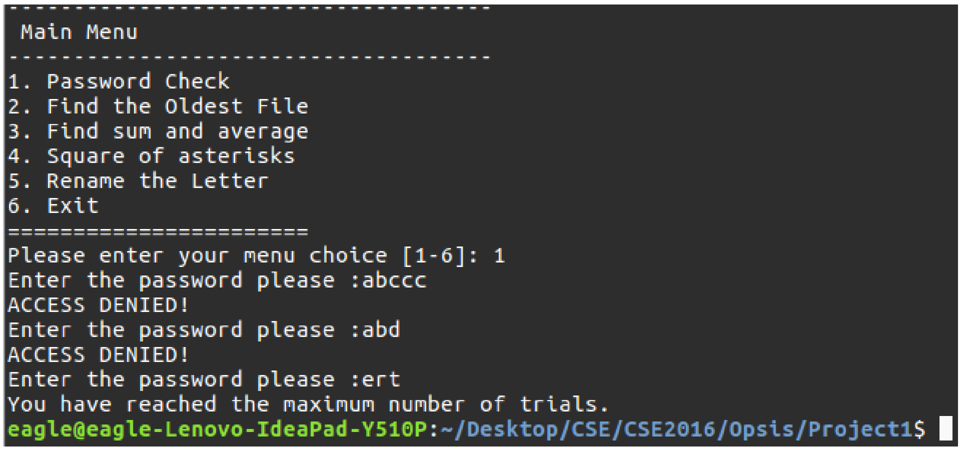
In this shell script project , a program was created that checks the password , shows oldest file or link or directory in the current directory we work, sums and finds average of entered numbers ,draws rectangle that is outlined two columns or rows and writes a letters with entered name as an input.

As this shell script is mentioned with details , when this program is executed , first a menu screen will be appeared . To select your choice and be able to apply for all menu selection , a password that is "cse333" is required . If a wrong password is entered three times , the shell script will stop itself .After entering the correct password , no password is required anymore. After executing one of the menus , pressing any key to skip other options.Until "6" is entered , this shell script will be executed. When "6" is written , the program will be closed.



```
-----------------------------------
 Main Menu
-----------------------------------
1. Password Check
2. Find the Oldest File
3. Find sum and average
4. Square of asterisks
5. Rename the Letter
6. Exit
======================
Please enter your menu choice [1-6]: 1
Enter the password please :cse333
Access granted
To continue press any key
```

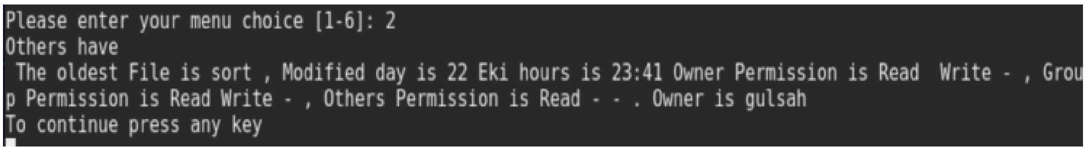
Figure1. Correct password is entered.



```
-------------------------------------
 Main Menu
-------------------------------------
1. Password Check
2. Find the Oldest File
3. Find sum and average
4. Square of asterisks
5. Rename the Letter
6. Exit
=======================
Please enter your menu choice [1-6]: 1
Enter the password please :abccc
ACCESS DENIED!
Enter the password please :abd
ACCESS DENIED!
Enter the password please :ert
You have reached the maximum number of trials.
eagle@eagle-Lenovo-IdeaPad-Y510P:~/Desktop/CSE/CSE2016/Opsis/Project1$
```

Figure2. Wrong password is entered three times.

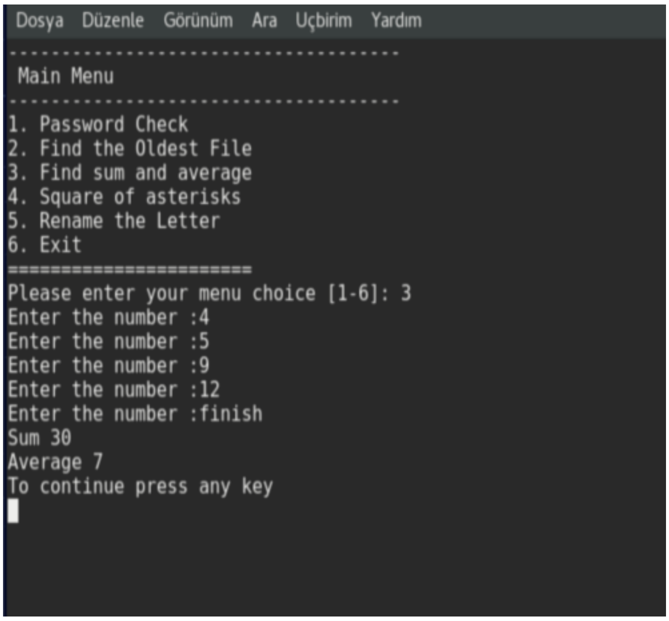
In the second choice , as it is said before , it shows properties of oldest file or directory or link in the current directory. Properties contain permissions , last modified date and time, owner . To see all these properties "ls -l -t | tail -1" command was used .



```
Please enter your menu choice [1-6]: 2
Others have
 The oldest File is sort , Modified day is 22 Eki hours is 23:41 Owner Permission is Read  Write - , Grou
p Permission is Read Write - , Others Permission is Read - - . Owner is gulsah
To continue press any key
```

Figure3. Menu 2 is selected.

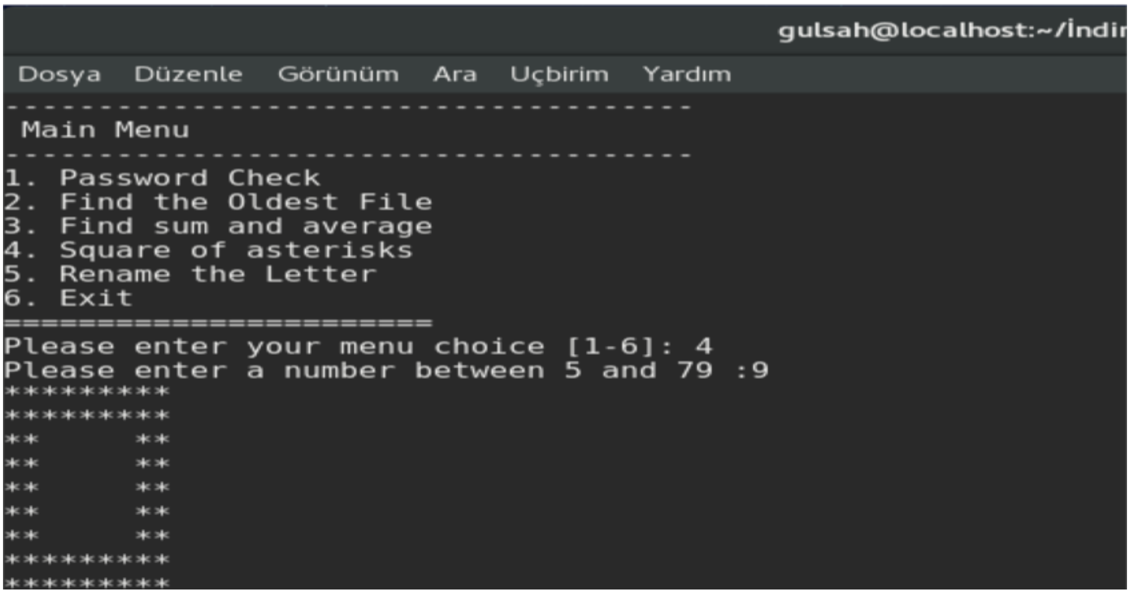
In the third option , it takes numbers as an input until the user writes “finish” .After the user enter “finish” ,this program will show sum and average of the numbers.



```
-----------------------------------
 Main Menu
-----------------------------------
1. Password Check
2. Find the Oldest File
3. Find sum and average
4. Square of asterisks
5. Rename the Letter
6. Exit
=======================
Please enter your menu choice [1-6]: 3
Enter the number :4
Enter the number :5
Enter the number :9
Enter the number :12
Enter the number :finish
Sum 30
Average 7
To continue press any key
```

Figure3. Output of the third option.

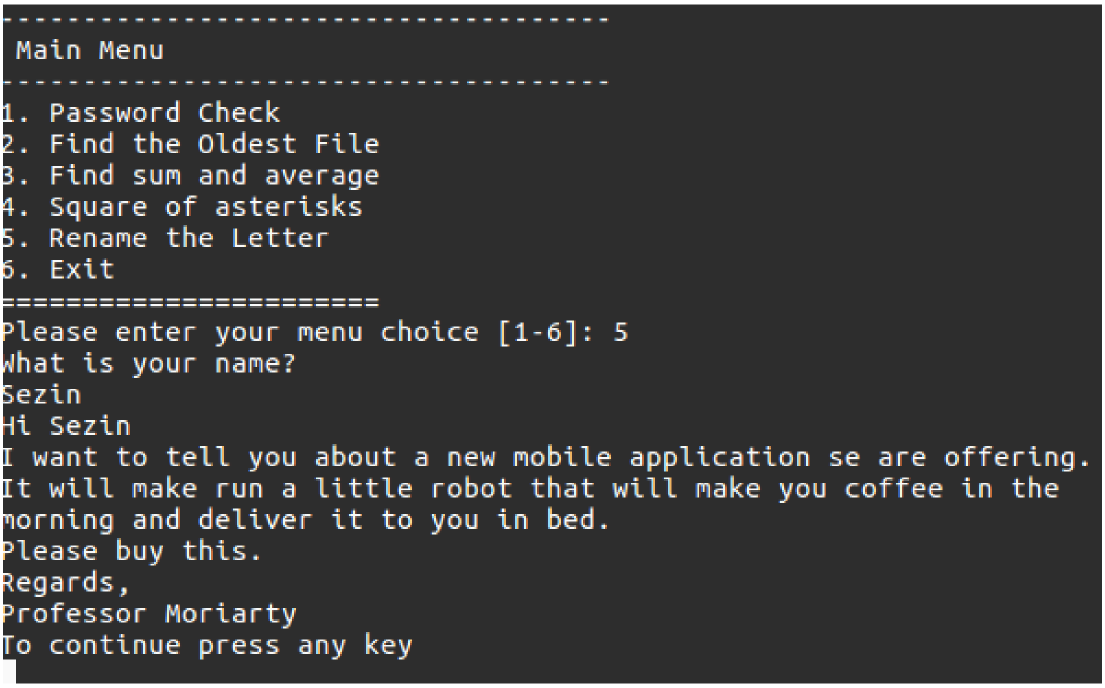
In the Menu 4 , a square with asterisk that is outlined two rows/columns and whose length is defined by the user the will be drawn . A length should be entered between 5 and 79. To provide this situation



```
-----------------------------------
 Main Menu
-----------------------------------
1. Password Check
2. Find the Oldest File
3. Find sum and average
4. Square of asterisks
5. Rename the Letter
6. Exit
=======================
Please enter your menu choice [1-6]: 4
Please enter a number between 5 and 79 :9
*********
*********
**     **
**     **
**     **
**     **
**     **
*********
*********
```

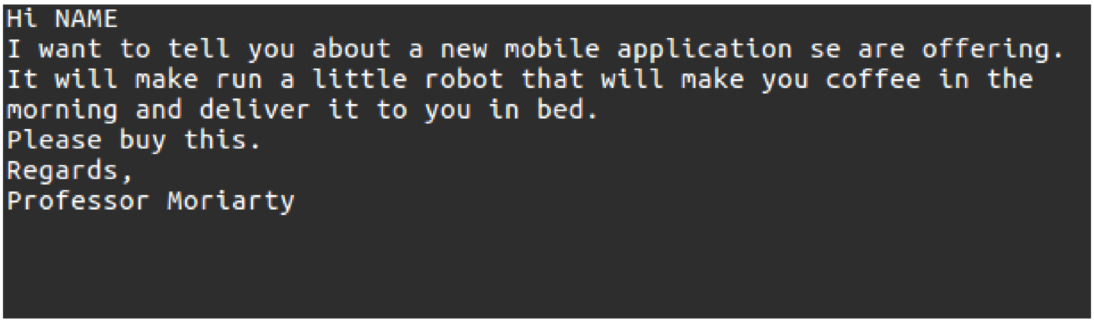
Figure4. Output of the Menu 4 , when the invalid length is entered and valid input is given.

In the last choice , a letter will be written in the “letter.txt” file. As the details are mentioned ,first of all , the user enter a name as input and the shell script takes the input and changes “NAME” word in the “letter.txt” file . Finally , it writes the letter on the terminal.”sed -i -e 's|NAME|'$lett'|g' letter.txt” command was used to update the line.



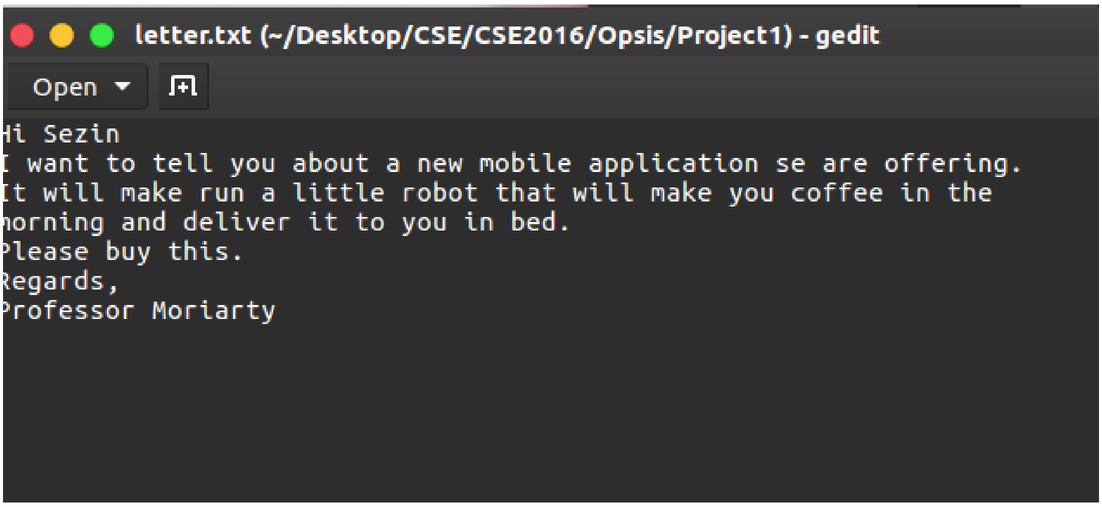
```
-------------------------------------
 Main Menu
-------------------------------------
1. Password Check
2. Find the Oldest File
3. Find sum and average
4. Square of asterisks
5. Rename the Letter
6. Exit
=======================
Please enter your menu choice [1-6]: 5
What is your name?
Sezin
Hi Sezin
I want to tell you about a new mobile application se are offering.
It will make run a little robot that will make you coffee in the
morning and deliver it to you in bed.
Please buy this.
Regards,
Professor Moriarty
To continue press any key
```

Figure5. The input is given in Menu 5.



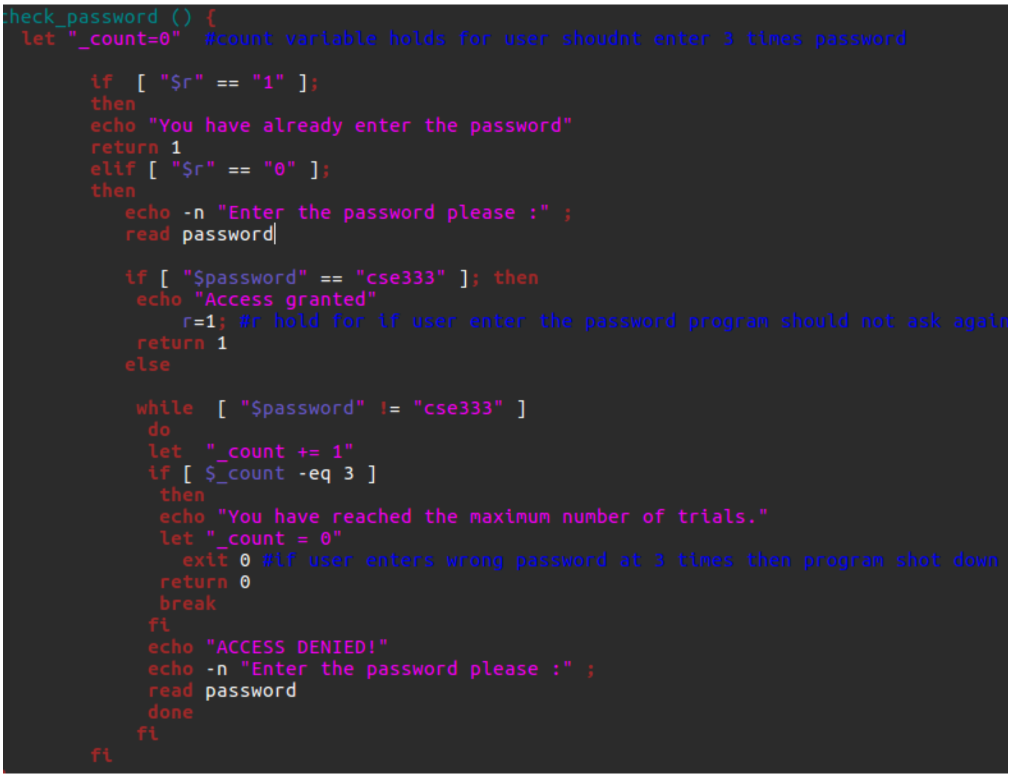
```
Hi NAME
I want to tell you about a new mobile application se are offering.
It will make run a little robot that will make you coffee in the
morning and deliver it to you in bed.
Please buy this.
Regards,
Professor Moriarty
```

Figure6. Before being entered the input from the user.



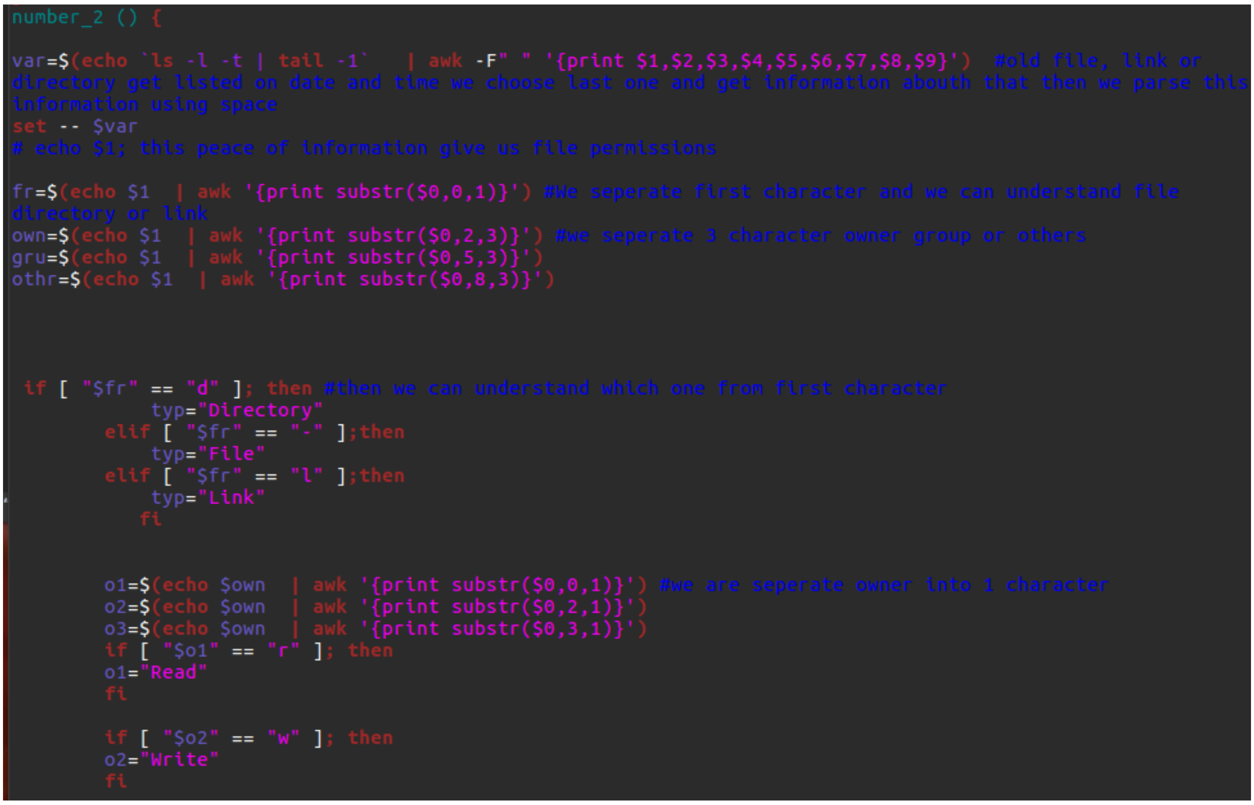
```
Hi Sezin
I want to tell you about a new mobile application se are offering.
It will make run a little robot that will make you coffee in the
morning and deliver it to you in bed.
Please buy this.
Regards,
Professor Moriarty
```

Figure7. After being given the input from the user.



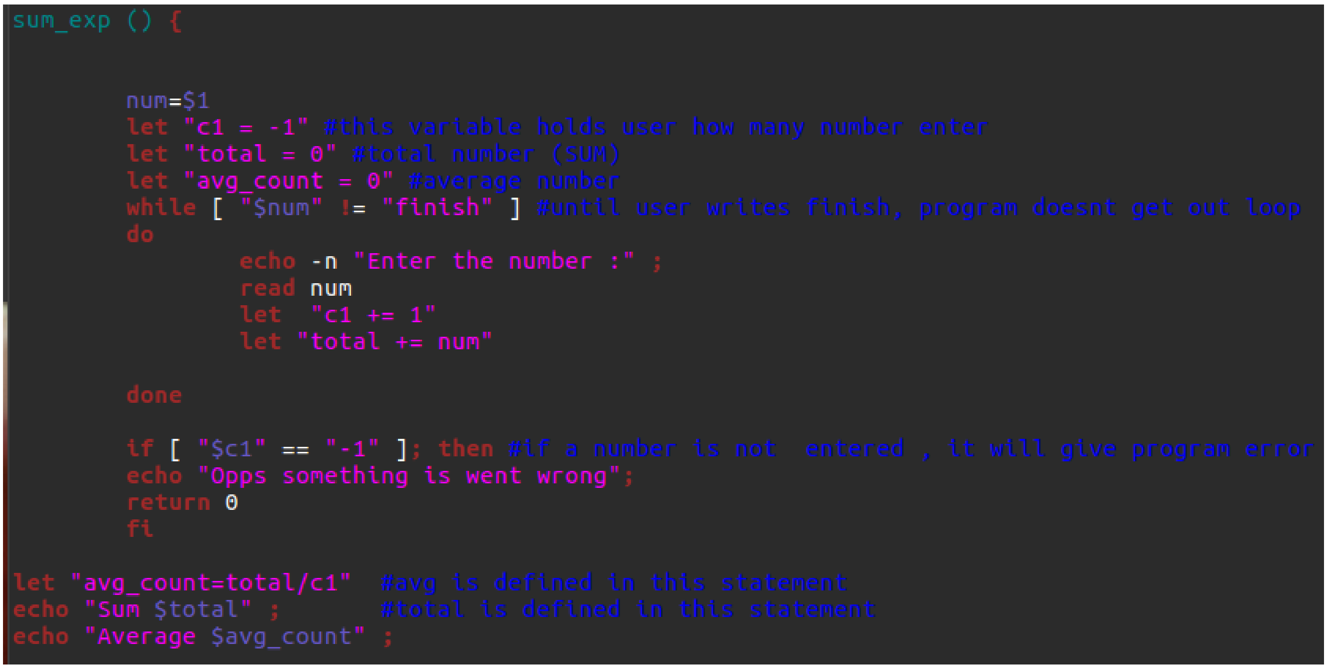
```
heck_password () {
 let "_count=0"  #count variable holds for user shoudnt enter 3 times password

       if  [ "$r" == "1" ];
       then
       echo "You have already enter the password"
       return 1
       elif [ "$r" == "0" ];
       then
          echo -n "Enter the password please :" ;
          read password

          if [ "$password" == "cse333" ]; then
           echo "Access granted"
               r=1; #r hold for if user enter the password program should not ask agai
           return 1
          else

           while  [ "$password" != "cse333" ]
            do
            let  "_count += 1"
            if [ $_count -eq 3 ]
             then
             echo "You have reached the maximum number of trials."
             let "_count = 0"
               exit 0 #if user enters wrong password at 3 times then program shot down
             return 0
             break
            fi
            echo "ACCESS DENIED!"
            echo -n "Enter the password please :" ;
            read password
            done
           fi
       fi
```

Source code of Menu 1.



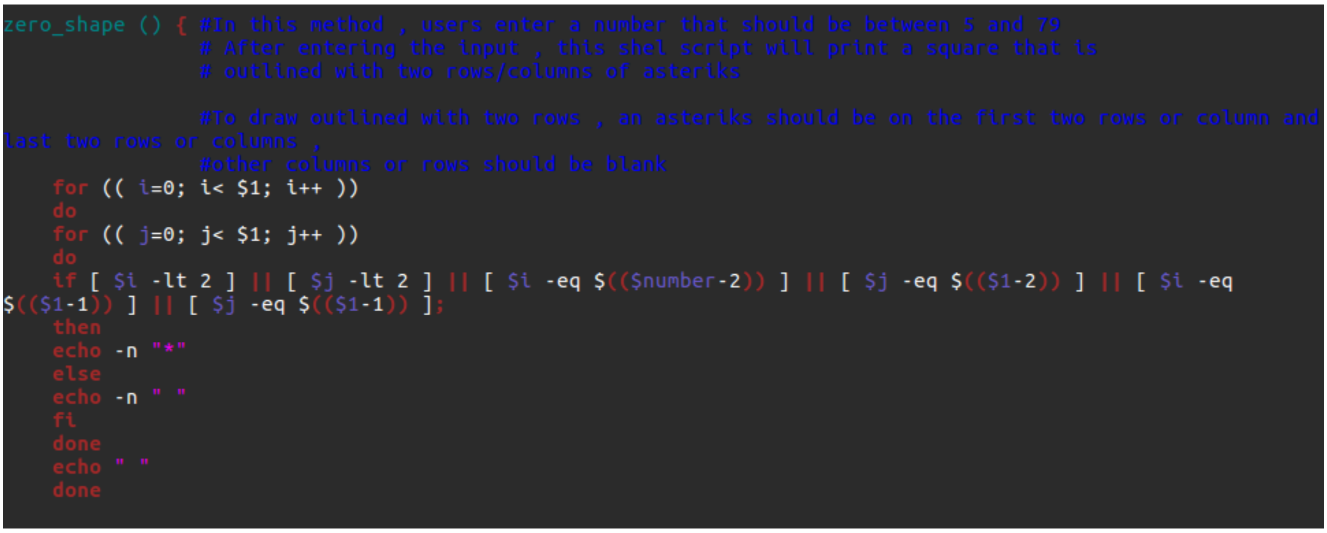
```
number_2 () {

var=$(echo `ls -l -t | tail -1`   | awk -F" " '{print $1,$2,$3,$4,$5,$6,$7,$8,$9}')  #old file, link or
directory get listed on date and time we choose last one and get information abouth that then we parse this
information using space
set -- $var
# echo $1; this peace of information give us file permissions

fr=$(echo $1  | awk '{print substr($0,0,1)}') #We seperate first character and we can understand file
directory or link
own=$(echo $1  | awk '{print substr($0,2,3)}') #we seperate 3 character owner group or others
gru=$(echo $1  | awk '{print substr($0,5,3)}')
othr=$(echo $1  | awk '{print substr($0,8,3)}')



 if [ "$fr" == "d" ]; then #then we can understand which one from first character
            typ="Directory"
        elif [ "$fr" == "-" ];then
            typ="File"
        elif [ "$fr" == "l" ];then
            typ="Link"
           fi


        o1=$(echo $own  | awk '{print substr($0,0,1)}') #we are seperate owner into 1 character
        o2=$(echo $own  | awk '{print substr($0,2,1)}')
        o3=$(echo $own  | awk '{print substr($0,3,1)}')
        if [ "$o1" == "r" ]; then
        o1="Read"
        fi

        if [ "$o2" == "w" ]; then
        o2="Write"
        fi
```

Source code of Menu 2.



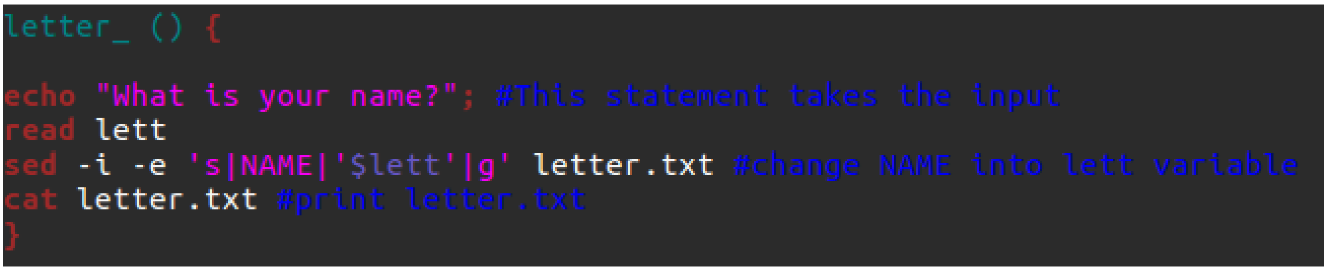
```
sum_exp () {


        num=$1
        let "c1 = -1" #this variable holds user how many number enter
        let "total = 0" #total number (SUM)
        let "avg_count = 0" #average number
        while [ "$num" != "finish" ] #until user writes finish, program doesnt get out loop
        do
                echo -n "Enter the number :" ;
                read num
                let  "c1 += 1"
                let "total += num"

        done

        if [ "$c1" == "-1" ]; then #if a number is not  entered , it will give program error
        echo "Opps something is went wrong";
        return 0
        fi

let "avg_count=total/c1"  #avg is defined in this statement
echo "Sum $total" ;       #total is defined in this statement
echo "Average $avg_count" ;
```

Source code of Menu 3.



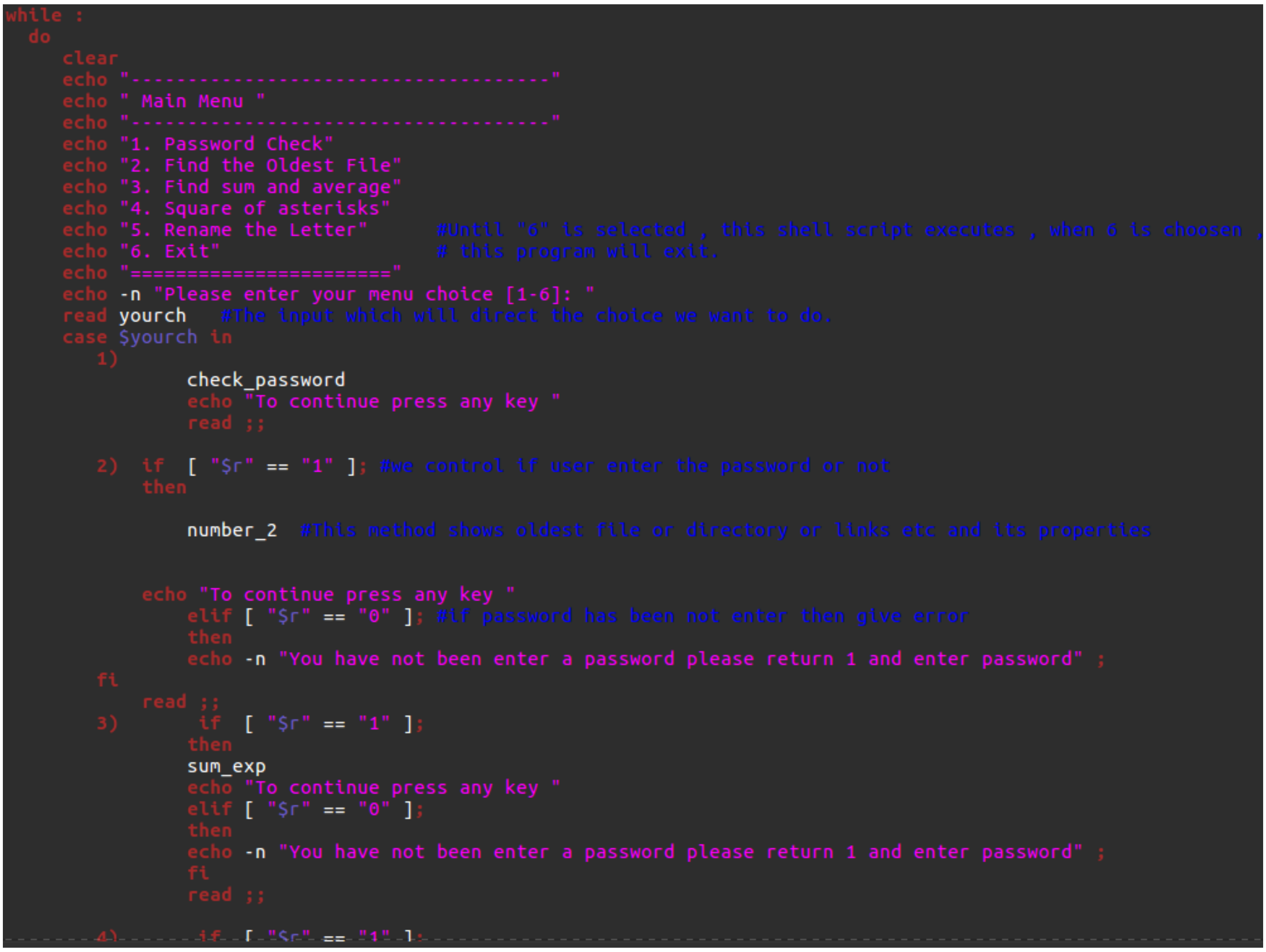
```
zero_shape () { #In this method , users enter a number that should be between 5 and 79
                # After entering the input , this shel script will print a square that is
                # outlined with two rows/columns of asteriks

                #To draw outlined with two rows , an asteriks should be on the first two rows or column and
last two rows or columns ,
                #other columns or rows should be blank
    for (( i=0; i< $1; i++ ))
    do
    for (( j=0; j< $1; j++ ))
    do
    if [ $i -lt 2 ] || [ $j -lt 2 ] || [ $i -eq $(($number-2)) ] || [ $j -eq $(($1-2)) ] || [ $i -eq
$(($1-1)) ] || [ $j -eq $(($1-1)) ];
    then
    echo -n "*"
    else
    echo -n " "
    fi
    done
    echo " "
    done
```

Source code of Menu 4.



```
letter_ () {

echo "What is your name?"; #This statement takes the input
read lett
sed -i -e 's|NAME|'$lett'|g' letter.txt #change NAME into lett variable
cat letter.txt #print letter.txt
}
```

Source code of Menu 5.



```
while :
  do
     clear
     echo "-------------------------------------"
     echo " Main Menu "
     echo "-------------------------------------"
     echo "1. Password Check"
     echo "2. Find the Oldest File"
     echo "3. Find sum and average"
     echo "4. Square of asterisks"
     echo "5. Rename the Letter"      #Until "6" is selected , this shell script executes , when 6 is choosen ,
     echo "6. Exit"                   # this program will exit.
     echo "======================="
     echo -n "Please enter your menu choice [1-6]: "
     read yourch   #The input which will direct the choice we want to do.
     case $yourch in
        1)
                check_password
                echo "To continue press any key "
                read ;;

        2)  if  [ "$r" == "1" ]; #we control if user enter the password or not
            then

                number_2  #This method shows oldest file or directory or links etc and its properties


            echo "To continue press any key "
                elif [ "$r" == "0" ]; #if password has been not enter then give error
                then
                echo -n "You have not been enter a password please return 1 and enter password" ;
        fi
            read ;;
        3)       if  [ "$r" == "1" ];
                then
                sum_exp
                echo "To continue press any key "
                elif [ "$r" == "0" ];
                then
                echo -n "You have not been enter a password please return 1 and enter password" ;
                fi
                read ;;

        4)       if  [ "$r" == "1" ];
```

Source code of Main Menu Screen.

SEZIN   GÜMÜŞ   150113841
GÜLŞAH YILMAZ 150113854