Faculty of Engineering & Technology Electrical & Computer Engineering Department

ENCS3130: Linux Laboratory

Shell Scripting Project – Medical Test Management System

Prepared by:

Taleed Mahmoud Hamadneh	1220006
Qasim Nidal Batrawi	1220204

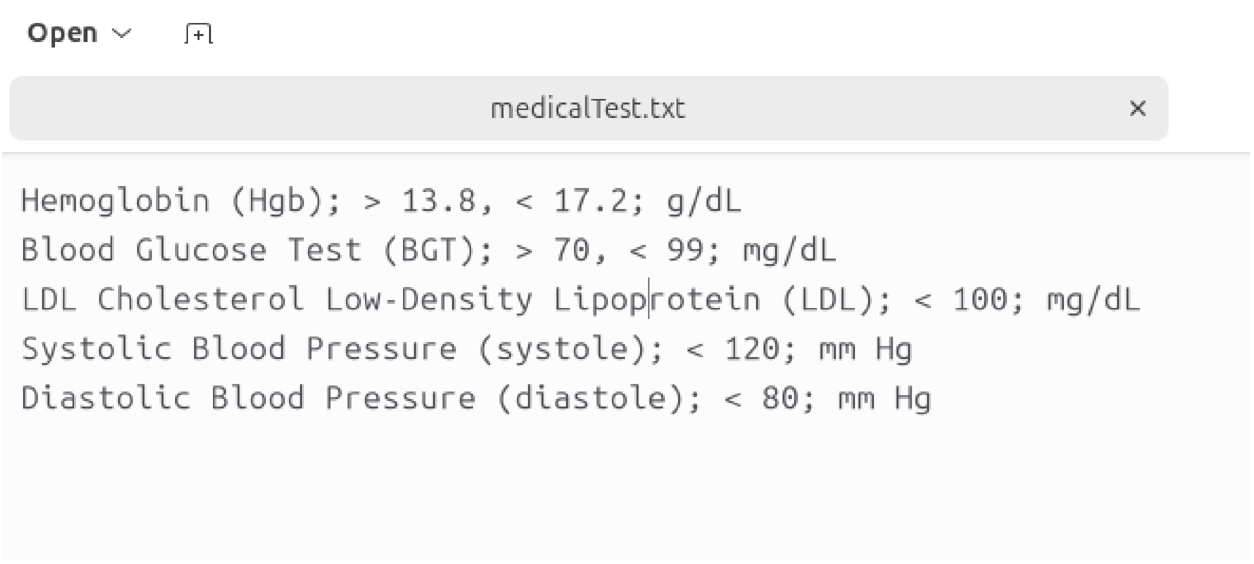
Instructor: Dr. Aziz Qaroush

TA: Ahed Mafarjeh

Section: 2

Date: 9 Aug 2024

medicalTest file data:

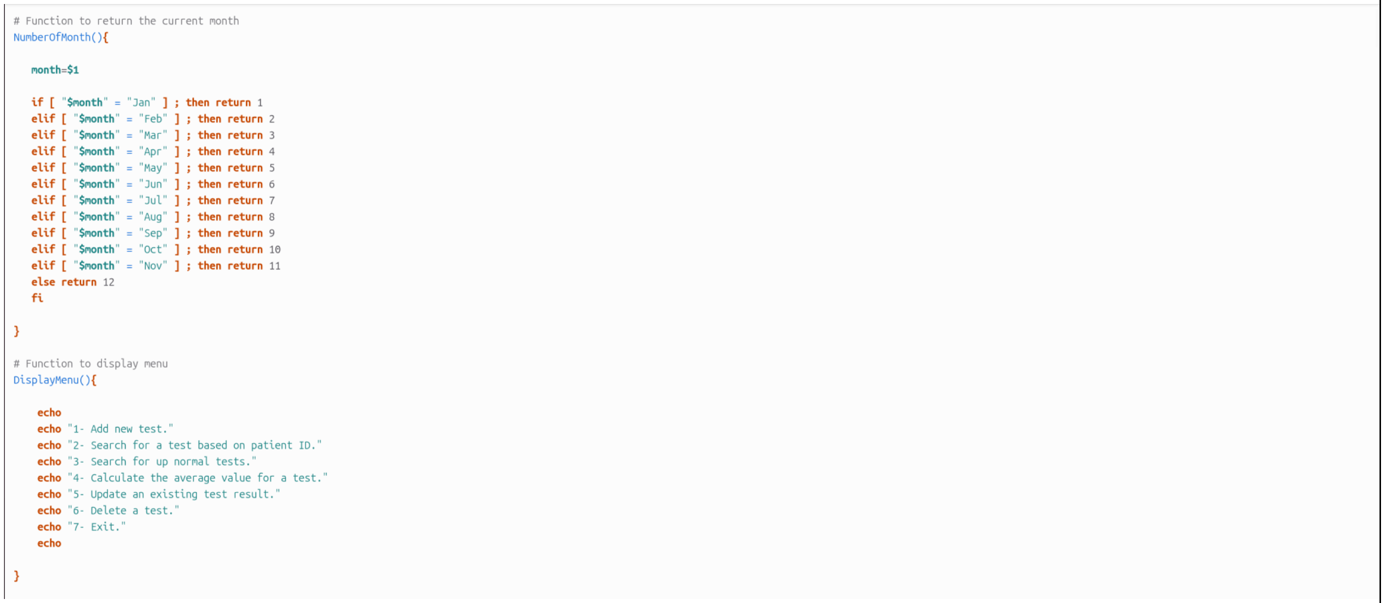


```
Hemoglobin (Hgb); > 13.8, < 17.2; g/dL
Blood Glucose Test (BGT); > 70, < 99; mg/dL
LDL Cholesterol Low-Density Lipoprotein (LDL); < 100; mg/dL
Systolic Blood Pressure (systole); < 120; mm Hg
Diastolic Blood Pressure (diastole); < 80; mm Hg
```

medicalRecord file

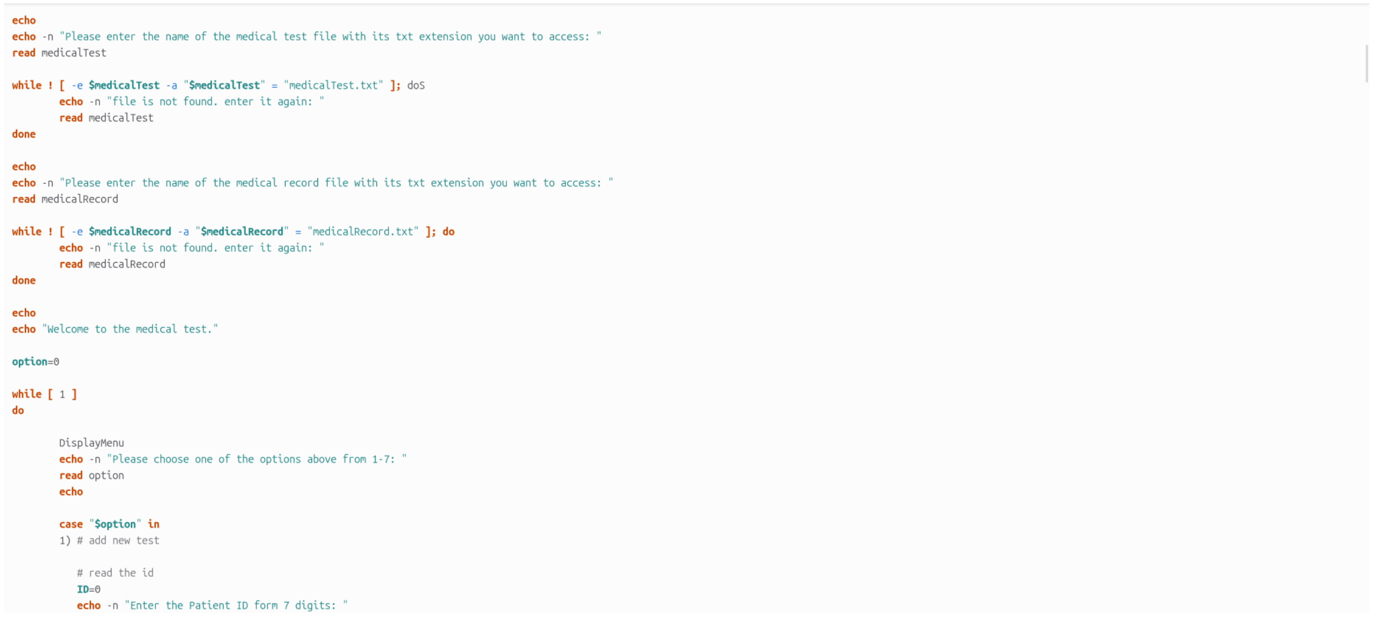
The file is currently empty

Shell Script file code:



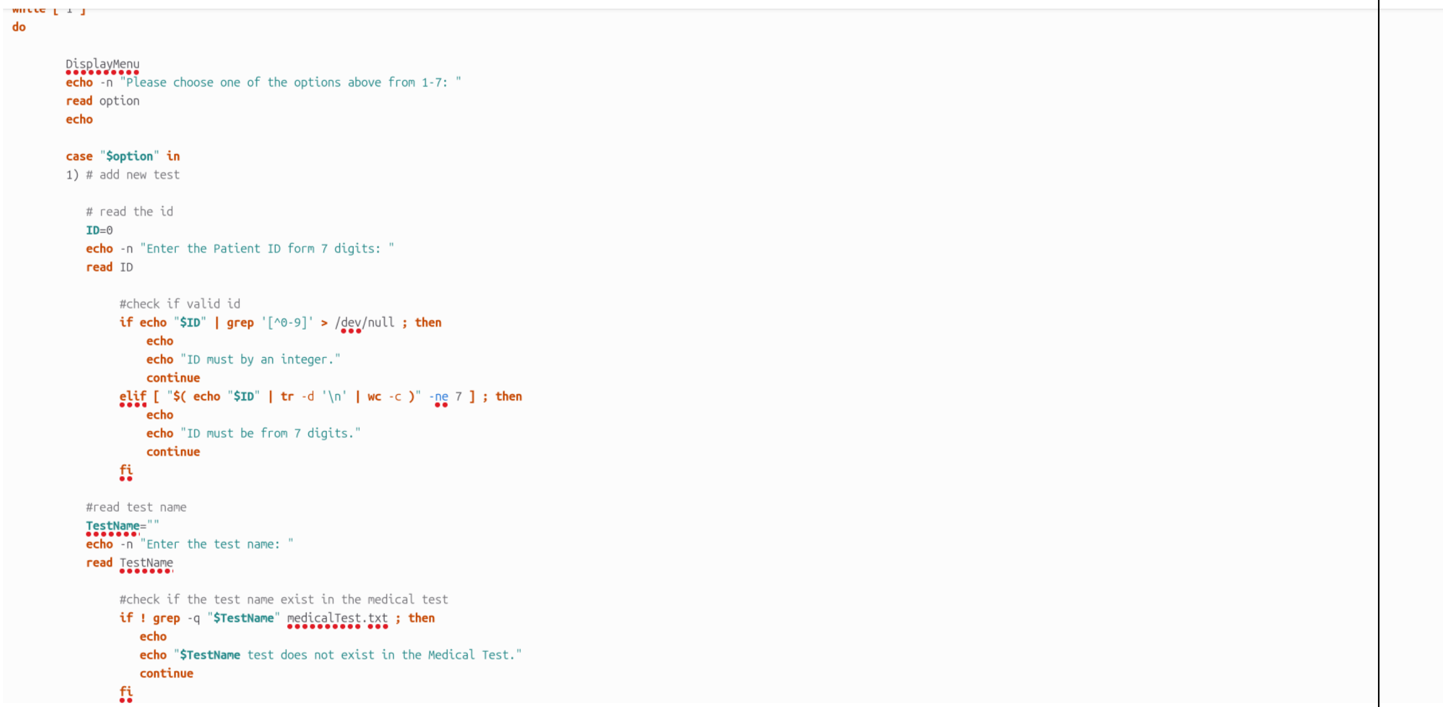
```
# Function to return the current month
NumberOfMonth(){

   month=$1

   if [ "$month" = "Jan" ] ; then return 1
   elif [ "$month" = "Feb" ] ; then return 2
   elif [ "$month" = "Mar" ] ; then return 3
   elif [ "$month" = "Apr" ] ; then return 4
   elif [ "$month" = "May" ] ; then return 5
   elif [ "$month" = "Jun" ] ; then return 6
   elif [ "$month" = "Jul" ] ; then return 7
   elif [ "$month" = "Aug" ] ; then return 8
   elif [ "$month" = "Sep" ] ; then return 9
   elif [ "$month" = "Oct" ] ; then return 10
   elif [ "$month" = "Nov" ] ; then return 11
   else return 12
   fi

}

# Function to display menu
DisplayMenu(){

    echo
    echo "1- Add new test."
    echo "2- Search for a test based on patient ID."
    echo "3- Search for up normal tests."
    echo "4- Calculate the average value for a test."
    echo "5- Update an existing test result."
    echo "6- Delete a test."
    echo "7- Exit."
    echo

}
```

Figure 1



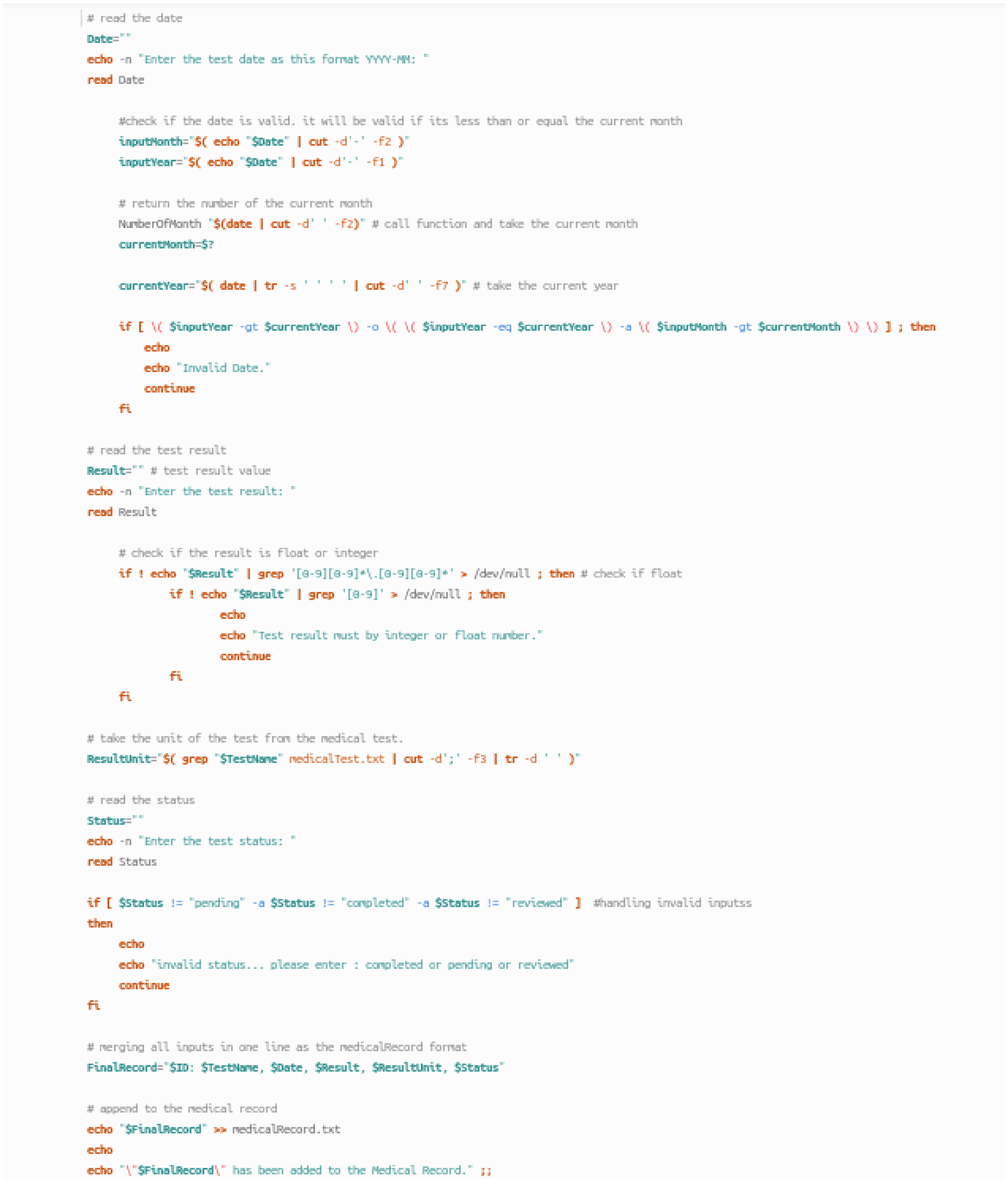
```
echo
echo -n "Please enter the name of the medical test file with its txt extension you want to access: "
read medicalTest

while ! [ -e $medicalTest -a "$medicalTest" = "medicalTest.txt" ]; doS
        echo -n "file is not found. enter it again: "
        read medicalTest
done

echo
echo -n "Please enter the name of the medical record file with its txt extension you want to access: "
read medicalRecord

while ! [ -e $medicalRecord -a "$medicalRecord" = "medicalRecord.txt" ]; do
        echo -n "file is not found. enter it again: "
        read medicalRecord
done

echo
echo "Welcome to the medical test."

option=0

while [ 1 ]
do

        DisplayMenu
        echo -n "Please choose one of the options above from 1-7: "
        read option
        echo

        case "$option" in
        1) # add new test

           # read the id
           ID=0
           echo -n "Enter the Patient ID form 7 digits: "
```

Figure 2



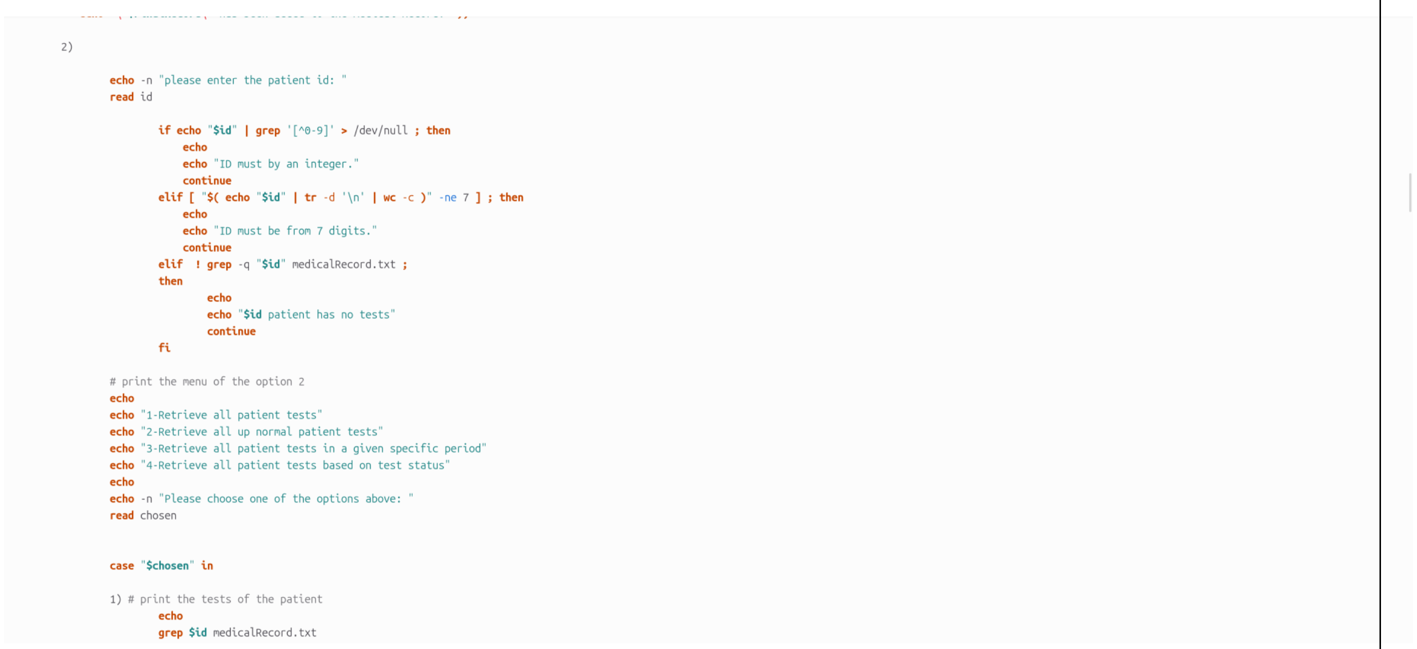
```
while [ 1 ]
do

        DisplayMenu
        echo -n "Please choose one of the options above from 1-7: "
        read option
        echo

        case "$option" in
        1) # add new test

           # read the id
           ID=0
           echo -n "Enter the Patient ID form 7 digits: "
           read ID

                #check if valid id
                if echo "$ID" | grep '[^0-9]' > /dev/null ; then
                    echo
                    echo "ID must by an integer."
                    continue
                elif [ "$( echo "$ID" | tr -d '\n' | wc -c )" -ne 7 ] ; then
                    echo
                    echo "ID must be from 7 digits."
                    continue
                fi

           #read test name
           TestName=""
           echo -n "Enter the test name: "
           read TestName

                #check if the test name exist in the medical test
                if ! grep -q "$TestName" medicalTest.txt ; then
                   echo
                   echo "$TestName test does not exist in the Medical Test."
                   continue
                fi
```

Figure 3



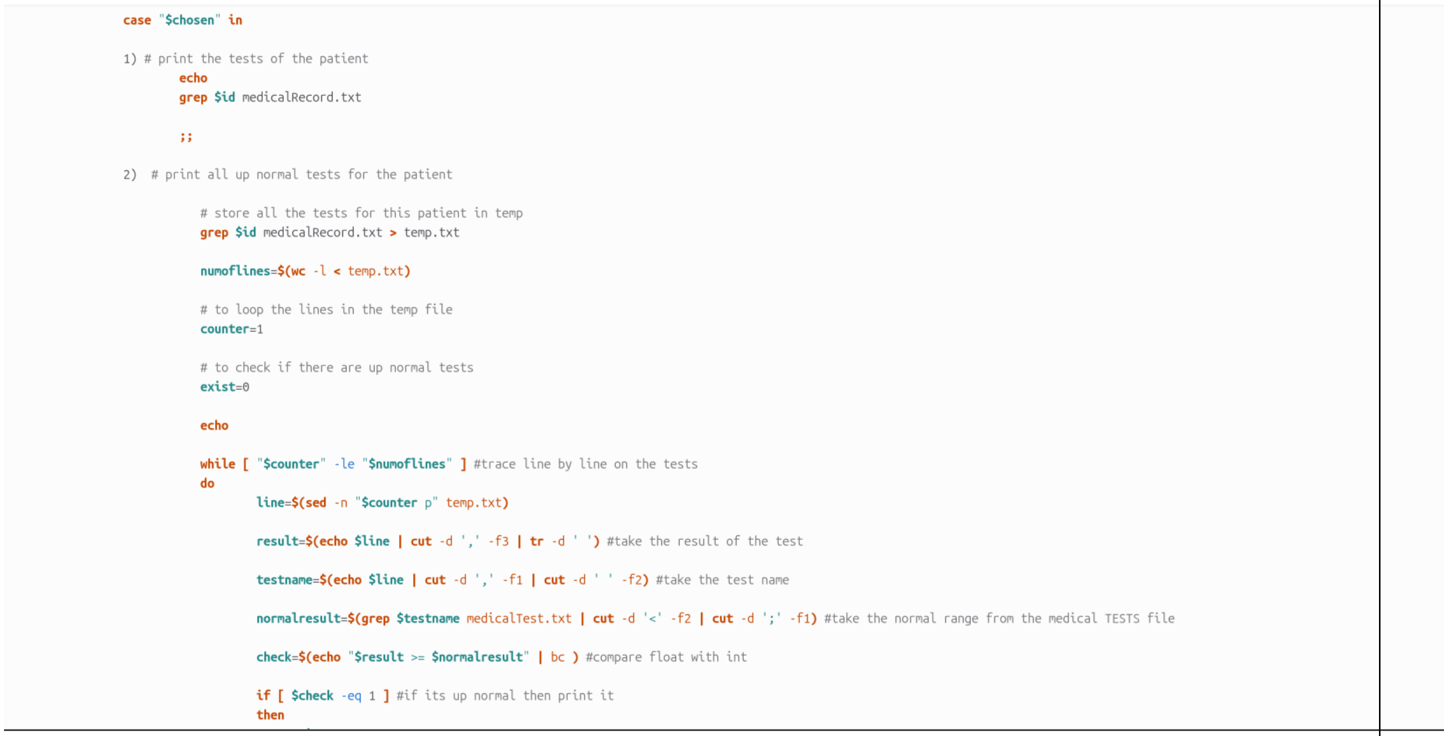
```bash
# read the date
Date=""
echo -n "Enter the test date as this format YYYY-MM: "
read Date

     #check if the date is valid. it will be valid if its less than or equal the current month
     inputMonth="$( echo "$Date" | cut -d'-' -f2 )"
     inputYear="$( echo "$Date" | cut -d'-' -f1 )"

     # return the number of the current month
     NumberOfMonth "$(date | cut -d' ' -f2)" # call function and take the current month
     currentMonth=$?

     currentYear="$( date | tr -s ' ' ' ' | cut -d' ' -f7 )" # take the current year

     if [ \( $inputYear -gt $currentYear \) -o \( \( $inputYear -eq $currentYear \) -a \( $inputMonth -gt $currentMonth \) \) ] ; then
         echo
         echo "Invalid Date."
         continue
     fi

# read the test result
Result="" # test result value
echo -n "Enter the test result: "
read Result

     # check if the result is float or integer
     if ! echo "$Result" | grep '[0-9][0-9]*\.[0-9][0-9]*' > /dev/null ; then # check if float
             if ! echo "$Result" | grep '[0-9]' > /dev/null ; then
                     echo
                     echo "Test result must by integer or float number."
                     continue
             fi
     fi

# take the unit of the test from the medical test.
ResultUnit="$( grep "$TestName" medicalTest.txt | cut -d';' -f3 | tr -d ' ' )"

# read the status
Status=""
echo -n "Enter the test status: "
read Status

if [ $Status != "pending" -a $Status != "completed" -a $Status != "reviewed" ]  #handling invalid inputss
then
     echo
     echo "invalid status... please enter : completed or pending or reviewed"
     continue
fi

# merging all inputs in one line as the medicalRecord format
FinalRecord="$ID: $TestName, $Date, $Result, $ResultUnit, $Status"

# append to the medical record
echo "$FinalRecord" >> medicalRecord.txt
echo
echo "\"$FinalRecord\" has been added to the Medical Record." ;;
```

Figure 4



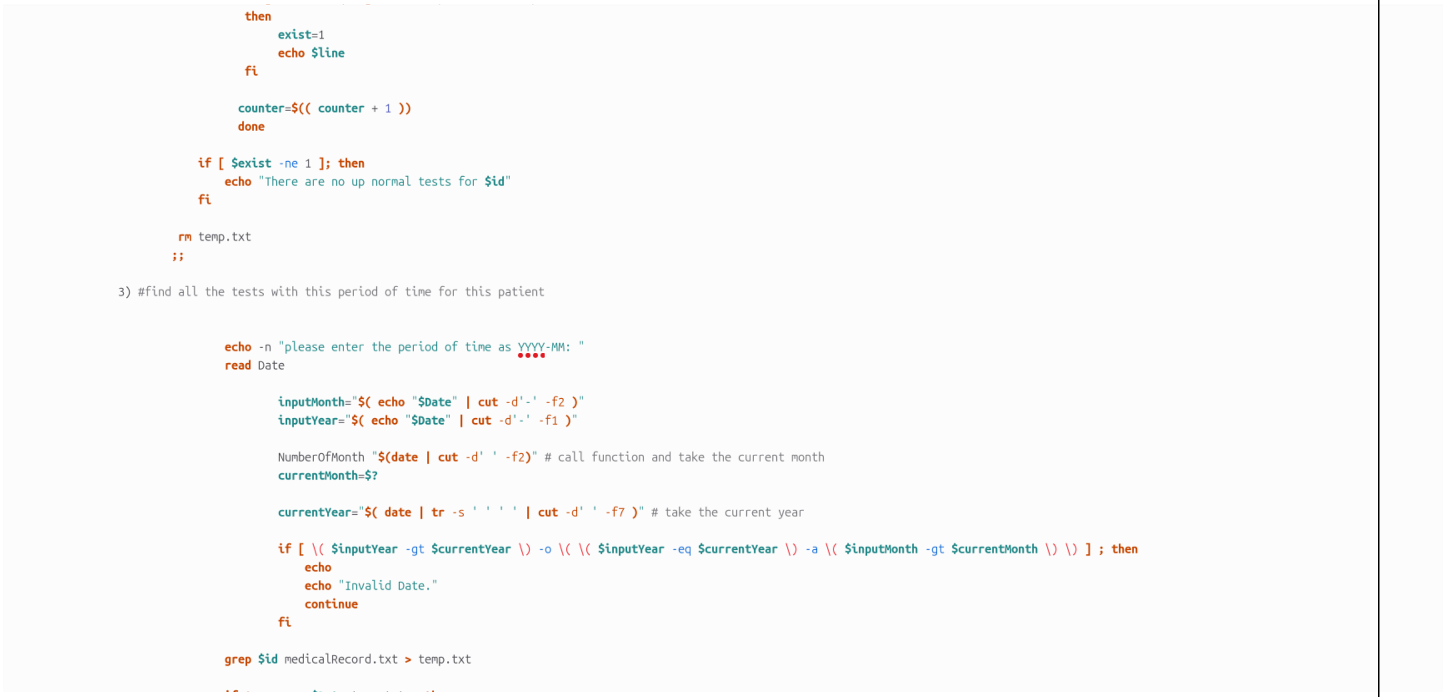
```
2)

        echo -n "please enter the patient id: "
        read id

                if echo "$id" | grep '[^0-9]' > /dev/null ; then
                    echo
                    echo "ID must by an integer."
                    continue
                elif [ "$( echo "$id" | tr -d '\n' | wc -c )" -ne 7 ] ; then
                    echo
                    echo "ID must be from 7 digits."
                    continue
                elif  ! grep -q "$id" medicalRecord.txt ;
                then
                        echo
                        echo "$id patient has no tests"
                        continue
                fi

        # print the menu of the option 2
        echo
        echo "1-Retrieve all patient tests"
        echo "2-Retrieve all up normal patient tests"
        echo "3-Retrieve all patient tests in a given specific period"
        echo "4-Retrieve all patient tests based on test status"
        echo
        echo -n "Please choose one of the options above: "
        read chosen


        case "$chosen" in

        1) # print the tests of the patient
                echo
                grep $id medicalRecord.txt
```

Figure 5



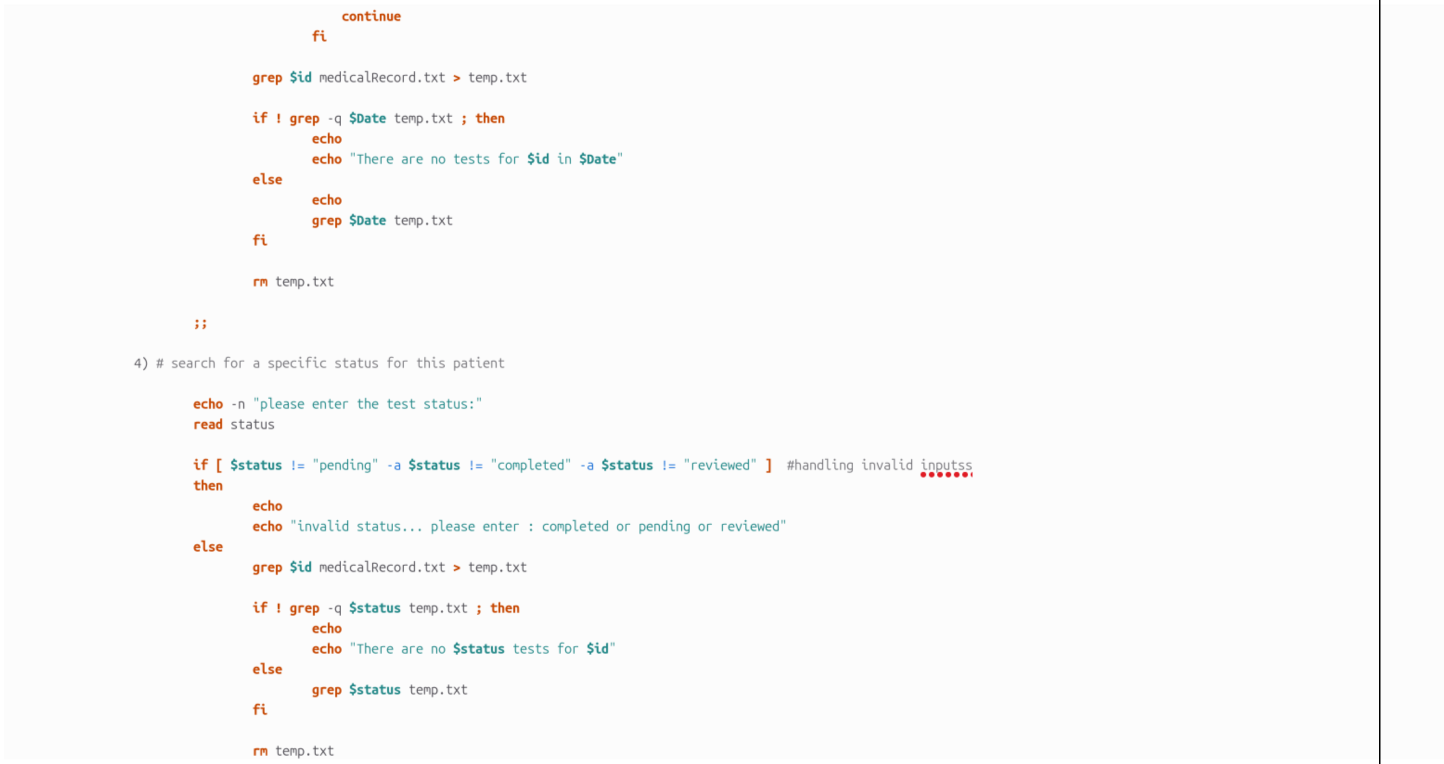
```
        case "$chosen" in

        1) # print the tests of the patient
                echo
                grep $id medicalRecord.txt

                ;;

        2)  # print all up normal tests for the patient

                  # store all the tests for this patient in temp
                  grep $id medicalRecord.txt > temp.txt

                  numoflines=$(wc -l < temp.txt)

                  # to loop the lines in the temp file
                  counter=1

                  # to check if there are up normal tests
                  exist=0

                  echo

                  while [ "$counter" -le "$numoflines" ] #trace line by line on the tests
                  do
                          line=$(sed -n "$counter p" temp.txt)

                          result=$(echo $line | cut -d ',' -f3 | tr -d ' ') #take the result of the test

                          testname=$(echo $line | cut -d ',' -f1 | cut -d ' ' -f2) #take the test name

                          normalresult=$(grep $testname medicalTest.txt | cut -d '<' -f2 | cut -d ';' -f1) #take the normal range from the medical TESTS file

                          check=$(echo "$result >= $normalresult" | bc ) #compare float with int

                          if [ $check -eq 1 ] #if its up normal then print it
                          then
```

Figure 6



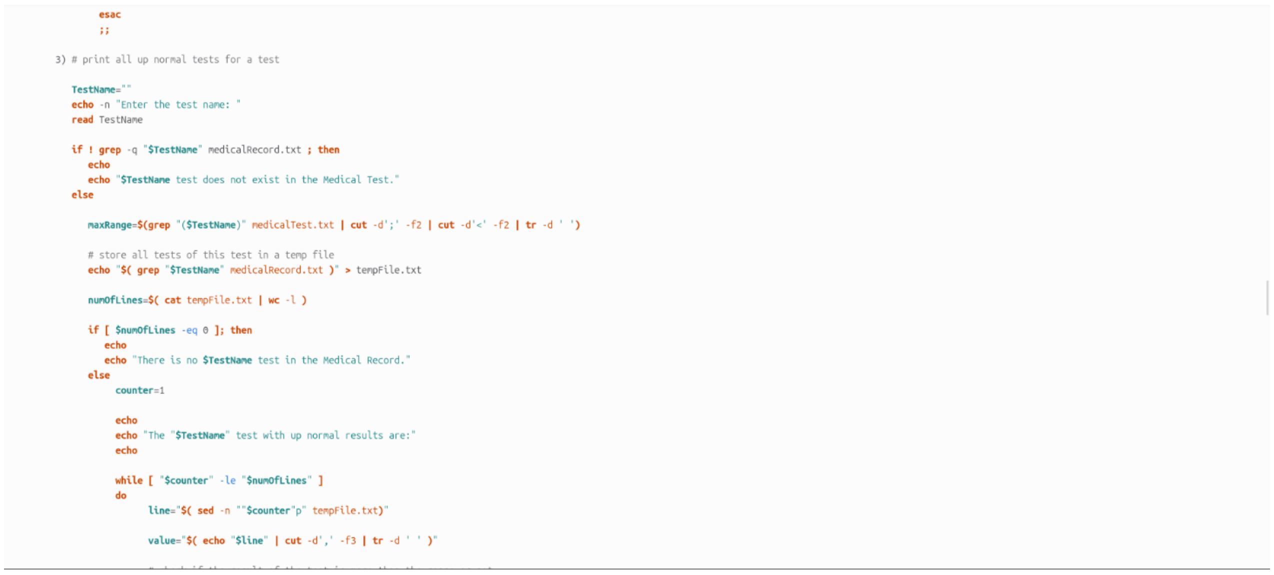
```
                   then
                        exist=1
                        echo $line
                   fi

                  counter=$(( counter + 1 ))
                  done

            if [ $exist -ne 1 ]; then
                echo "There are no up normal tests for $id"
            fi

         rm temp.txt
        ;;

3) #find all the tests with this period of time for this patient


                echo -n "please enter the period of time as YYYY-MM: "
                read Date

                        inputMonth="$( echo "$Date" | cut -d'-' -f2 )"
                        inputYear="$( echo "$Date" | cut -d'-' -f1 )"

                        NumberOfMonth "$(date | cut -d' ' -f2)" # call function and take the current month
                        currentMonth=$?

                        currentYear="$( date | tr -s ' ' ' ' | cut -d' ' -f7 )" # take the current year

                        if [ \( $inputYear -gt $currentYear \) -o \( \( $inputYear -eq $currentYear \) -a \( $inputMonth -gt $currentMonth \) \) ] ; then
                            echo
                            echo "Invalid Date."
                            continue
                        fi

                grep $id medicalRecord.txt > temp.txt
```

Figure 7



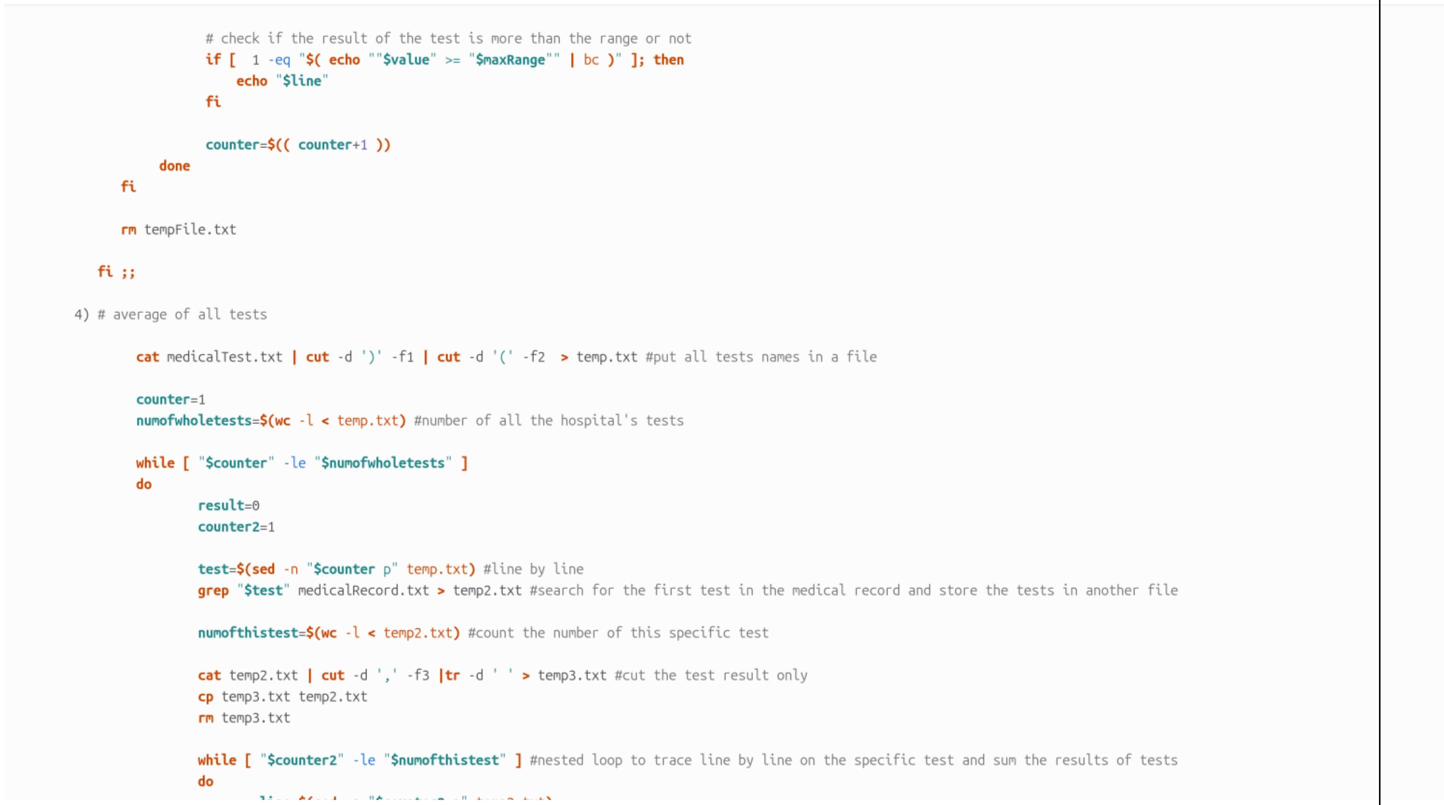
```
                            continue
                        fi

                grep $id medicalRecord.txt > temp.txt

                if ! grep -q $Date temp.txt ; then
                        echo
                        echo "There are no tests for $id in $Date"
                else
                        echo
                        grep $Date temp.txt
                fi

                rm temp.txt

        ;;

4) # search for a specific status for this patient

        echo -n "please enter the test status:"
        read status

        if [ $status != "pending" -a $status != "completed" -a $status != "reviewed" ]  #handling invalid inputss
        then
                echo
                echo "invalid status... please enter : completed or pending or reviewed"
        else
                grep $id medicalRecord.txt > temp.txt

                if ! grep -q $status temp.txt ; then
                        echo
                        echo "There are no $status tests for $id"
                else
                        grep $status temp.txt
                fi

                rm temp.txt
```

Figure 8



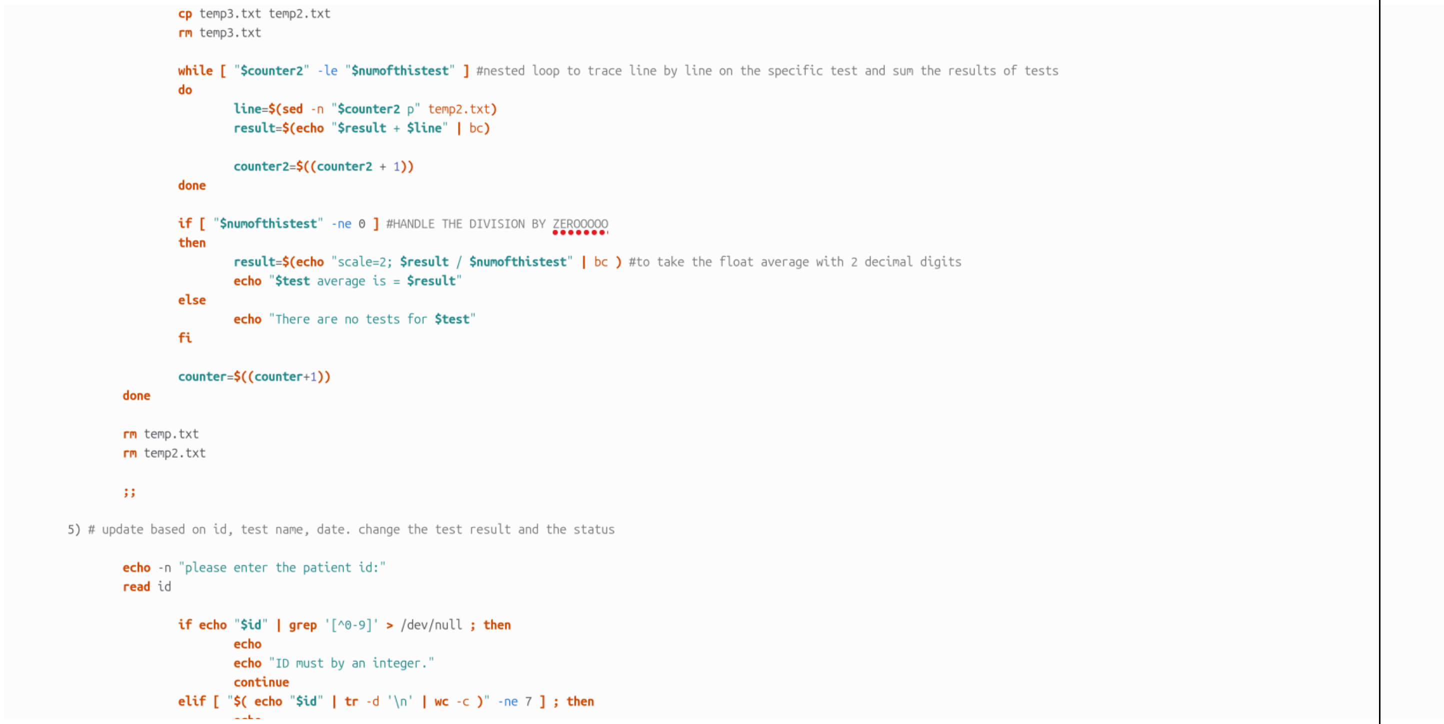
```
        esac
        ;;

3) # print all up normal tests for a test

   TestName=""
   echo -n "Enter the test name: "
   read TestName

   if ! grep -q "$TestName" medicalRecord.txt ; then
      echo
      echo "$TestName test does not exist in the Medical Test."
   else

      maxRange=$(grep "($TestName)" medicalTest.txt | cut -d';' -f2 | cut -d'<' -f2 | tr -d ' ')

      # store all tests of this test in a temp file
      echo "$( grep "$TestName" medicalRecord.txt )" > tempFile.txt

      numOfLines=$( cat tempFile.txt | wc -l )

      if [ $numOfLines -eq 0 ]; then
         echo
         echo "There is no $TestName test in the Medical Record."
      else
           counter=1

           echo
           echo "The "$TestName" test with up normal results are:"
           echo

           while [ "$counter" -le "$numOfLines" ]
           do
                 line="$( sed -n ""$counter"p" tempFile.txt)"

                 value="$( echo "$line" | cut -d',' -f3 | tr -d ' ' )"
```

Figure 9



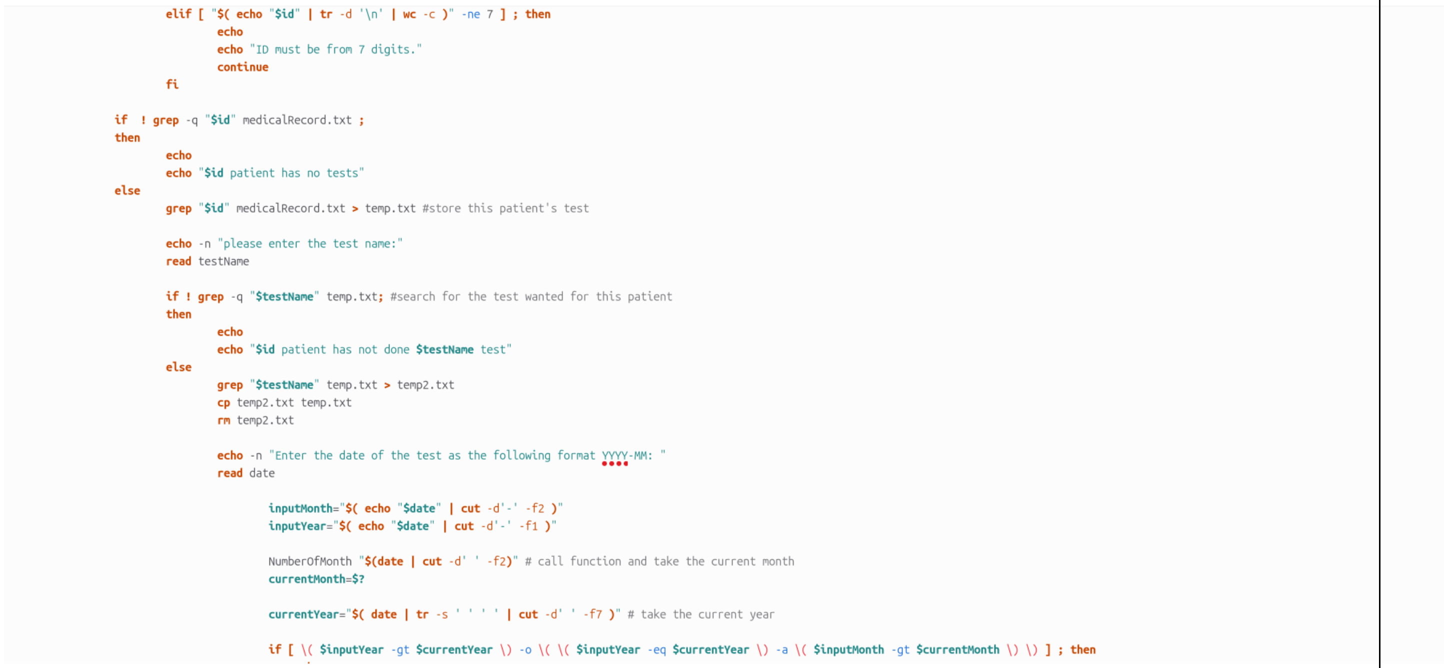
```
                    # check if the result of the test is more than the range or not
                    if [  1 -eq "$( echo ""$value" >= "$maxRange"" | bc )" ]; then
                        echo "$line"
                    fi

                    counter=$(( counter+1 ))
               done
          fi

          rm tempFile.txt

       fi ;;

    4) # average of all tests

            cat medicalTest.txt | cut -d ')' -f1 | cut -d '(' -f2  > temp.txt #put all tests names in a file

            counter=1
            numofwholetests=$(wc -l < temp.txt) #number of all the hospital's tests

            while [ "$counter" -le "$numofwholetests" ]
            do
                    result=0
                    counter2=1

                    test=$(sed -n "$counter p" temp.txt) #line by line
                    grep "$test" medicalRecord.txt > temp2.txt #search for the first test in the medical record and store the tests in another file

                    numofthistest=$(wc -l < temp2.txt) #count the number of this specific test

                    cat temp2.txt | cut -d ',' -f3 |tr -d ' ' > temp3.txt #cut the test result only
                    cp temp3.txt temp2.txt
                    rm temp3.txt

                    while [ "$counter2" -le "$numofthistest" ] #nested loop to trace line by line on the specific test and sum the results of tests
                    do
```

Figure 10



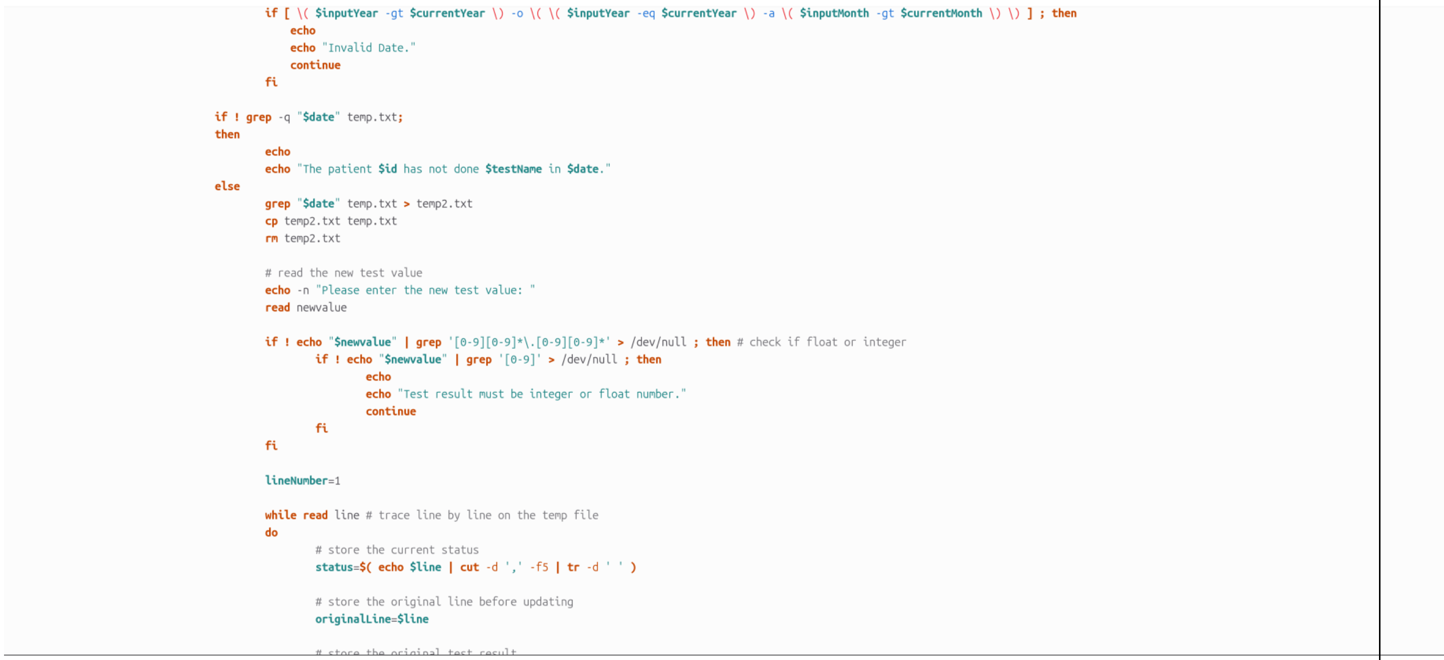
```
                cp temp3.txt temp2.txt
                rm temp3.txt

                while [ "$counter2" -le "$numofthistest" ] #nested loop to trace line by line on the specific test and sum the results of tests
                do
                        line=$(sed -n "$counter2 p" temp2.txt)
                        result=$(echo "$result + $line" | bc)

                        counter2=$((counter2 + 1))
                done

                if [ "$numofthistest" -ne 0 ] #HANDLE THE DIVISION BY ZEROOOOO
                then
                        result=$(echo "scale=2; $result / $numofthistest" | bc ) #to take the float average with 2 decimal digits
                        echo "$test average is = $result"
                else
                        echo "There are no tests for $test"
                fi

                counter=$((counter+1))
        done

        rm temp.txt
        rm temp2.txt

        ;;

5) # update based on id, test name, date. change the test result and the status

        echo -n "please enter the patient id:"
        read id

                if echo "$id" | grep '[^0-9]' > /dev/null ; then
                        echo
                        echo "ID must by an integer."
                        continue
                elif [ "$( echo "$id" | tr -d '\n' | wc -c )" -ne 7 ] ; then
```

Figure 11



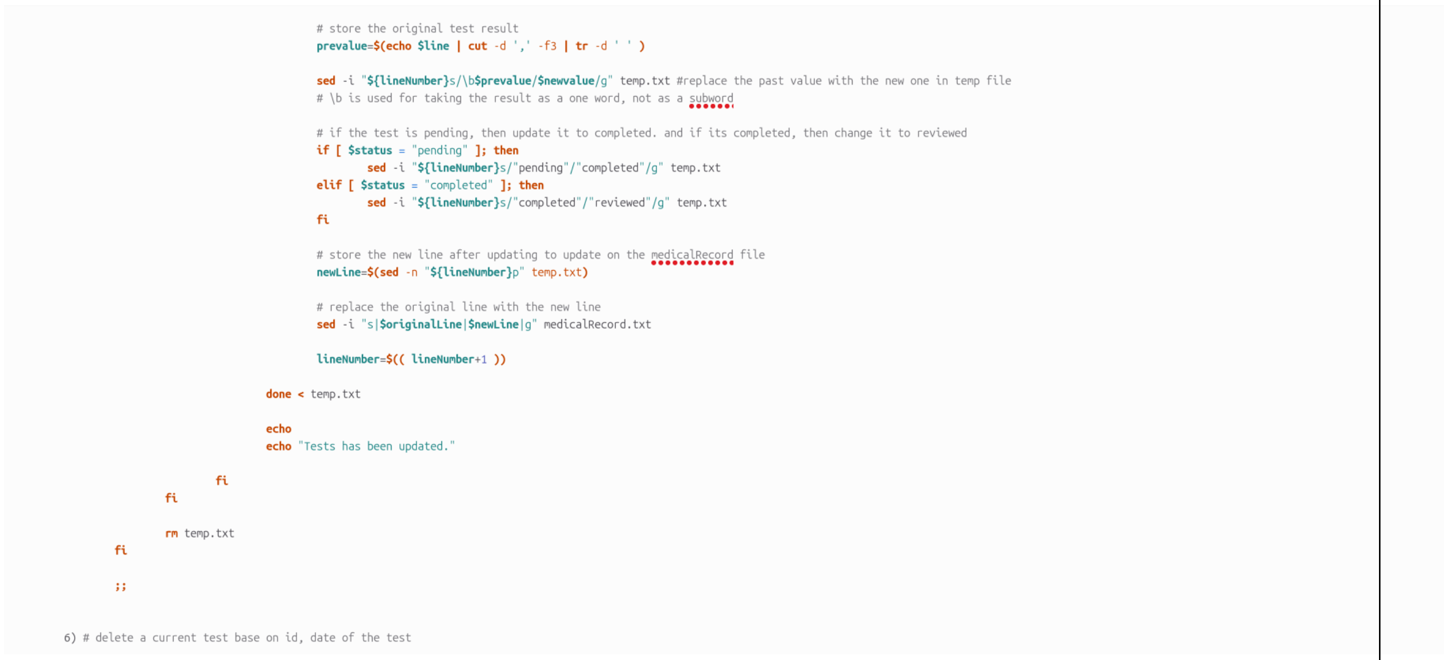
```
                elif [ "$( echo "$id" | tr -d '\n' | wc -c )" -ne 7 ] ; then
                        echo
                        echo "ID must be from 7 digits."
                        continue
                fi

        if  ! grep -q "$id" medicalRecord.txt ;
        then
                echo
                echo "$id patient has no tests"
        else
                grep "$id" medicalRecord.txt > temp.txt #store this patient's test

                echo -n "please enter the test name:"
                read testName

                if ! grep -q "$testName" temp.txt; #search for the test wanted for this patient
                then
                        echo
                        echo "$id patient has not done $testName test"
                else
                        grep "$testName" temp.txt > temp2.txt
                        cp temp2.txt temp.txt
                        rm temp2.txt

                        echo -n "Enter the date of the test as the following format YYYY-MM: "
                        read date

                                inputMonth="$( echo "$date" | cut -d'-' -f2 )"
                                inputYear="$( echo "$date" | cut -d'-' -f1 )"

                                NumberOfMonth "$(date | cut -d' ' -f2)" # call function and take the current month
                                currentMonth=$?

                                currentYear="$( date | tr -s ' ' ' ' | cut -d' ' -f7 )" # take the current year

                                if [ \( $inputYear -gt $currentYear \) -o \( \( $inputYear -eq $currentYear \) -a \( $inputMonth -gt $currentMonth \) \) ] ; then
```

Figure 12



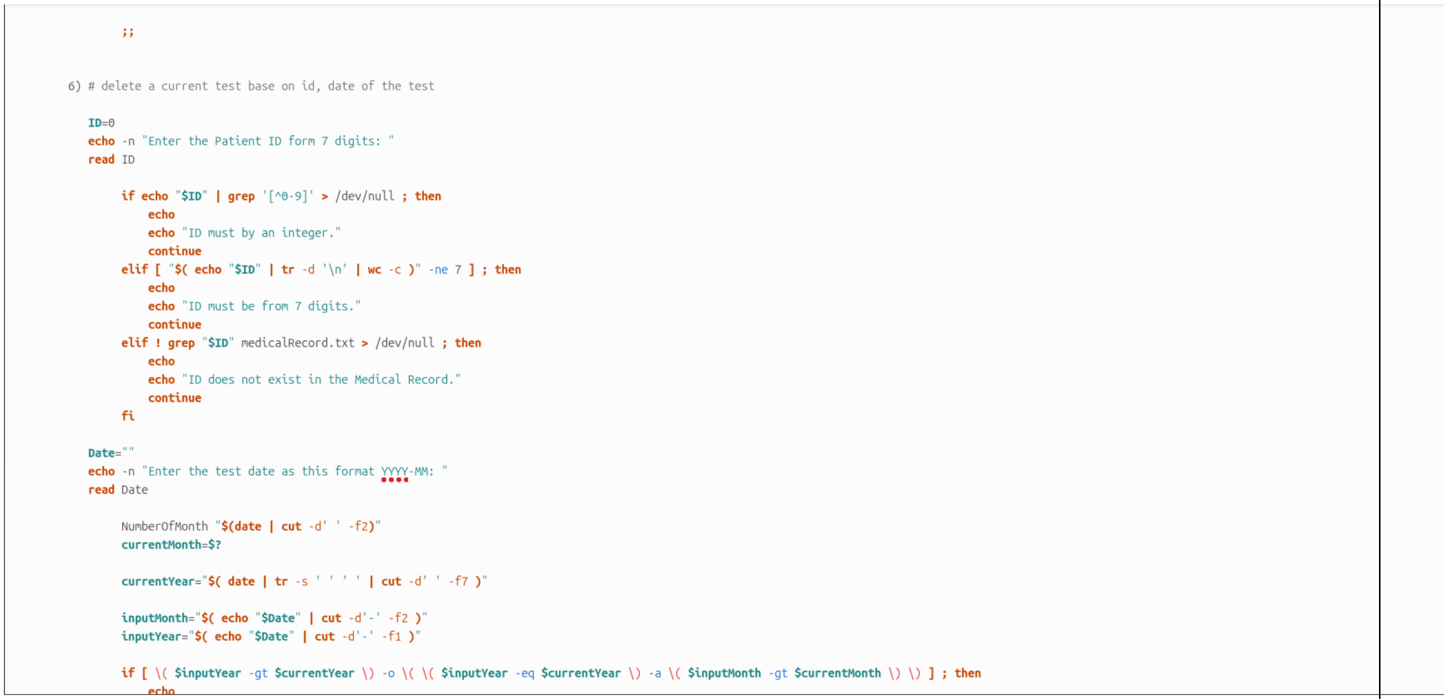
```
                if [ \( $inputYear -gt $currentYear \) -o \( \( $inputYear -eq $currentYear \) -a \( $inputMonth -gt $currentMonth \) \) ] ; then
                    echo
                    echo "Invalid Date."
                    continue
                fi

            if ! grep -q "$date" temp.txt;
            then
                echo
                echo "The patient $id has not done $testName in $date."
            else
                grep "$date" temp.txt > temp2.txt
                cp temp2.txt temp.txt
                rm temp2.txt

                # read the new test value
                echo -n "Please enter the new test value: "
                read newvalue

                if ! echo "$newvalue" | grep '[0-9][0-9]*\.[0-9][0-9]*' > /dev/null ; then # check if float or integer
                    if ! echo "$newvalue" | grep '[0-9]' > /dev/null ; then
                        echo
                        echo "Test result must be integer or float number."
                        continue
                    fi
                fi

                lineNumber=1

                while read line # trace line by line on the temp file
                do
                    # store the current status
                    status=$( echo $line | cut -d ',' -f5 | tr -d ' ' )

                    # store the original line before updating
                    originalLine=$line

                    # store the original test result
```

Figure 13



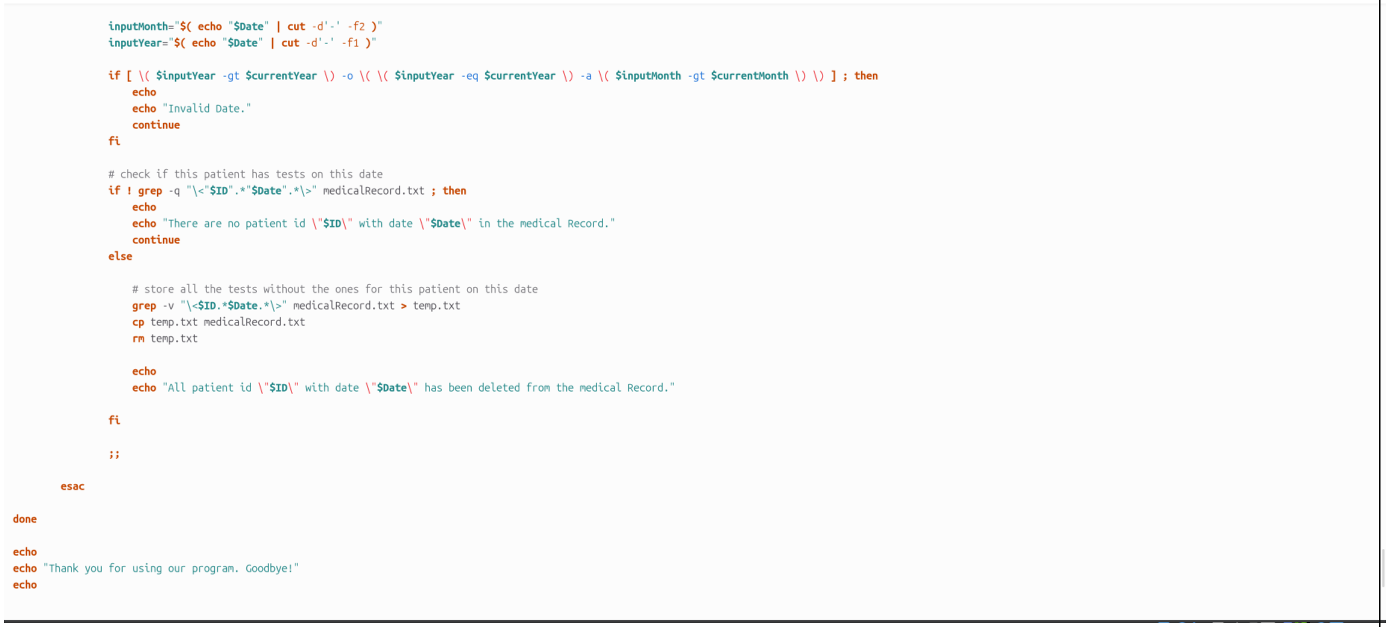
```
                    # store the original test result
                    prevalue=$(echo $line | cut -d ',' -f3 | tr -d ' ' )

                    sed -i "${lineNumber}s/\b$prevalue/$newvalue/g" temp.txt #replace the past value with the new one in temp file
                    # \b is used for taking the result as a one word, not as a subword

                    # if the test is pending, then update it to completed. and if its completed, then change it to reviewed
                    if [ $status = "pending" ]; then
                        sed -i "${lineNumber}s/"pending"/"completed"/g" temp.txt
                    elif [ $status = "completed" ]; then
                        sed -i "${lineNumber}s/"completed"/"reviewed"/g" temp.txt
                    fi

                    # store the new line after updating to update on the medicalRecord file
                    newLine=$(sed -n "${lineNumber}p" temp.txt)

                    # replace the original line with the new line
                    sed -i "s|$originalLine|$newLine|g" medicalRecord.txt

                    lineNumber=$(( lineNumber+1 ))

                done < temp.txt

                echo
                echo "Tests has been updated."

            fi
        fi

        rm temp.txt
    fi

    ;;


6) # delete a current test base on id, date of the test
```

Figure 14



```
            ;;


        6) # delete a current test base on id, date of the test

          ID=0
          echo -n "Enter the Patient ID form 7 digits: "
          read ID

               if echo "$ID" | grep '[^0-9]' > /dev/null ; then
                   echo
                   echo "ID must by an integer."
                   continue
               elif [ "$( echo "$ID" | tr -d '\n' | wc -c )" -ne 7 ] ; then
                   echo
                   echo "ID must be from 7 digits."
                   continue
               elif ! grep "$ID" medicalRecord.txt > /dev/null ; then
                   echo
                   echo "ID does not exist in the Medical Record."
                   continue
               fi

          Date=""
          echo -n "Enter the test date as this format YYYY-MM: "
          read Date

               NumberOfMonth "$(date | cut -d' ' -f2)"
               currentMonth=$?

               currentYear="$( date | tr -s ' ' ' ' | cut -d' ' -f7 )"

               inputMonth="$( echo "$Date" | cut -d'-' -f2 )"
               inputYear="$( echo "$Date" | cut -d'-' -f1 )"

               if [ \( $inputYear -gt $currentYear \) -o \( \( $inputYear -eq $currentYear \) -a \( $inputMonth -gt $currentMonth \) \) ] ; then
                   echo
```

Figure 15

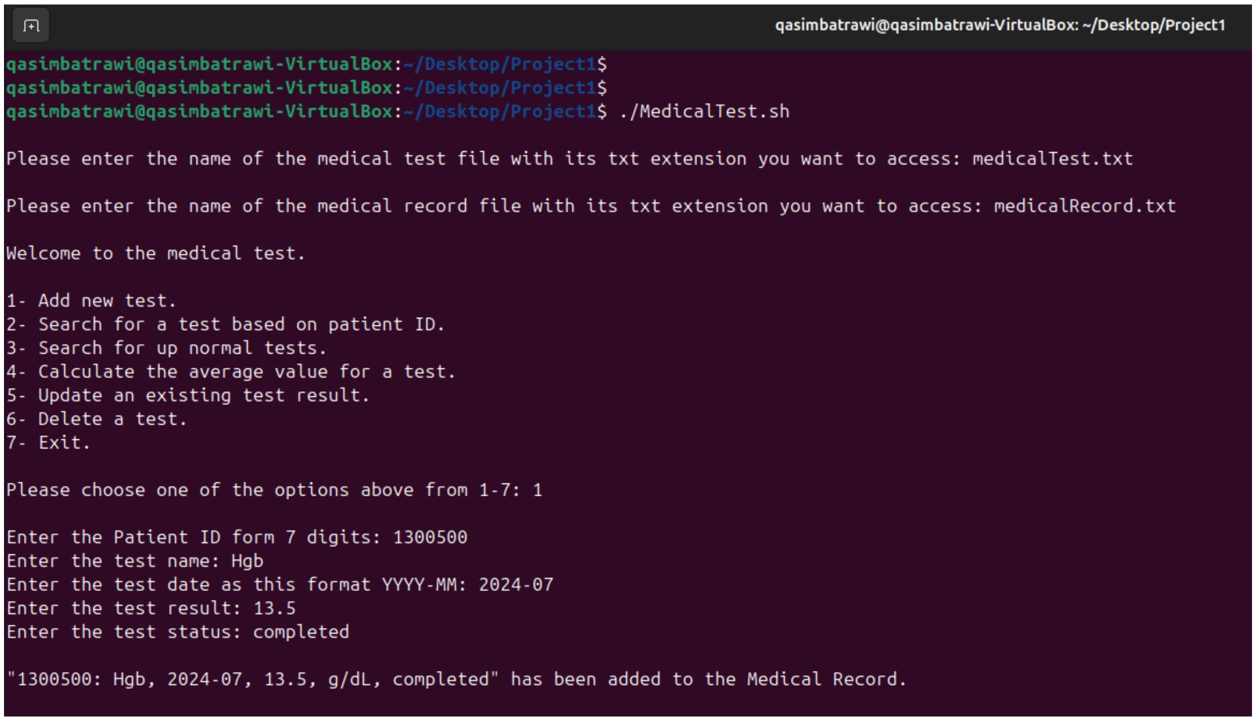


```
               inputMonth="$( echo "$Date" | cut -d'-' -f2 )"
               inputYear="$( echo "$Date" | cut -d'-' -f1 )"

               if [ \( $inputYear -gt $currentYear \) -o \( \( $inputYear -eq $currentYear \) -a \( $inputMonth -gt $currentMonth \) \) ] ; then
                   echo
                   echo "Invalid Date."
                   continue
               fi

               # check if this patient has tests on this date
               if ! grep -q "\<"$ID".*"$Date".*\>" medicalRecord.txt ; then
                   echo
                   echo "There are no patient id \"$ID\" with date \"$Date\" in the medical Record."
                   continue
               else

                   # store all the tests without the ones for this patient on this date
                   grep -v "\<$ID.*$Date.*\>" medicalRecord.txt > temp.txt
                   cp temp.txt medicalRecord.txt
                   rm temp.txt

                   echo
                   echo "All patient id \"$ID\" with date \"$Date\" has been deleted from the medical Record."

               fi

               ;;

           esac

       done

       echo
       echo "Thank you for using our program. Goodbye!"
       echo
```

Figure 16

Test Cases:

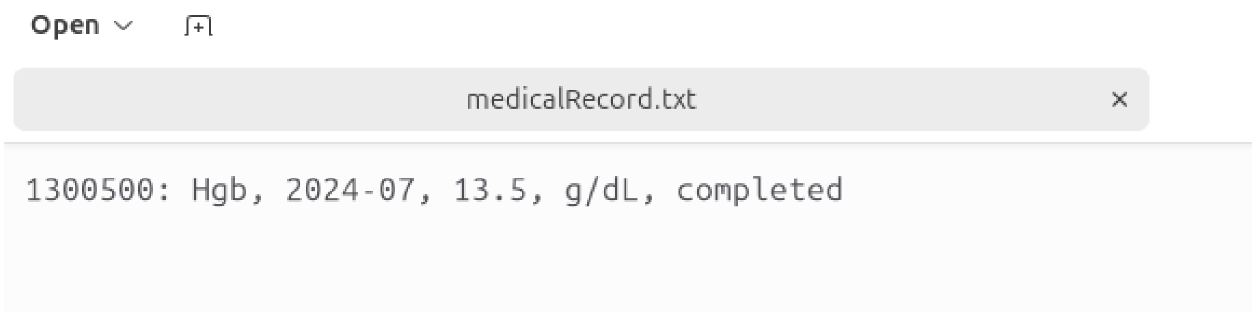
- Insert tests to the medical record file



qasimbatrawi@qasimbatrawi-VirtualBox: ~/Desktop/Project1

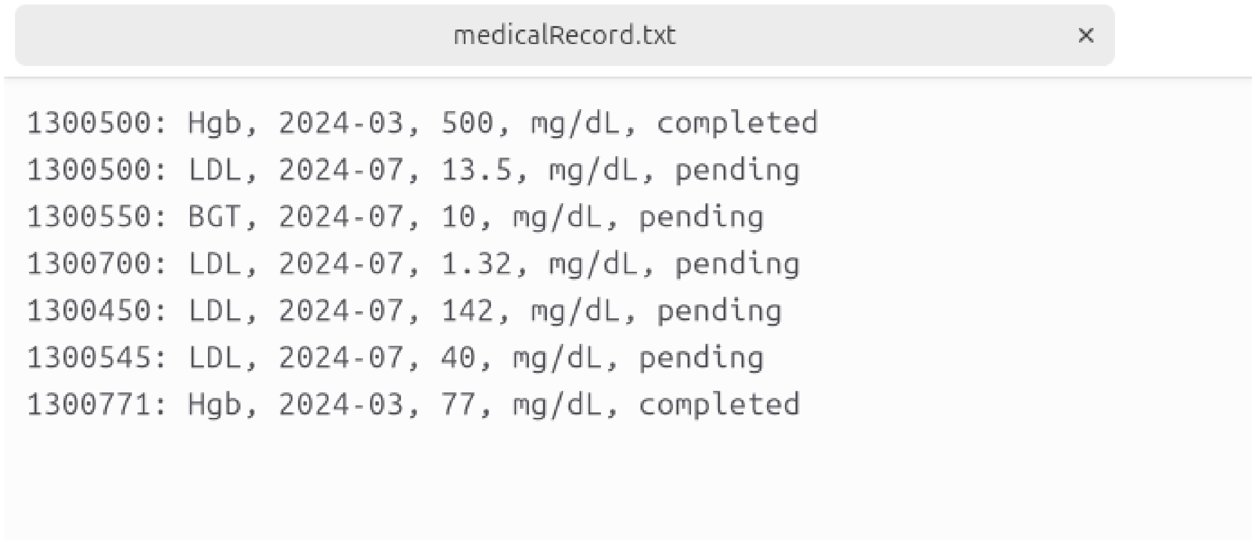
```
qasimbatrawi@qasimbatrawi-VirtualBox:~/Desktop/Project1$
qasimbatrawi@qasimbatrawi-VirtualBox:~/Desktop/Project1$
qasimbatrawi@qasimbatrawi-VirtualBox:~/Desktop/Project1$ ./MedicalTest.sh

Please enter the name of the medical test file with its txt extension you want to access: medicalTest.txt

Please enter the name of the medical record file with its txt extension you want to access: medicalRecord.txt

Welcome to the medical test.

1- Add new test.
2- Search for a test based on patient ID.
3- Search for up normal tests.
4- Calculate the average value for a test.
5- Update an existing test result.
6- Delete a test.
7- Exit.

Please choose one of the options above from 1-7: 1

Enter the Patient ID form 7 digits: 1300500
Enter the test name: Hgb
Enter the test date as this format YYYY-MM: 2024-07
Enter the test result: 13.5
Enter the test status: completed

"1300500: Hgb, 2024-07, 13.5, g/dL, completed" has been added to the Medical Record.
```

The file after inserting:



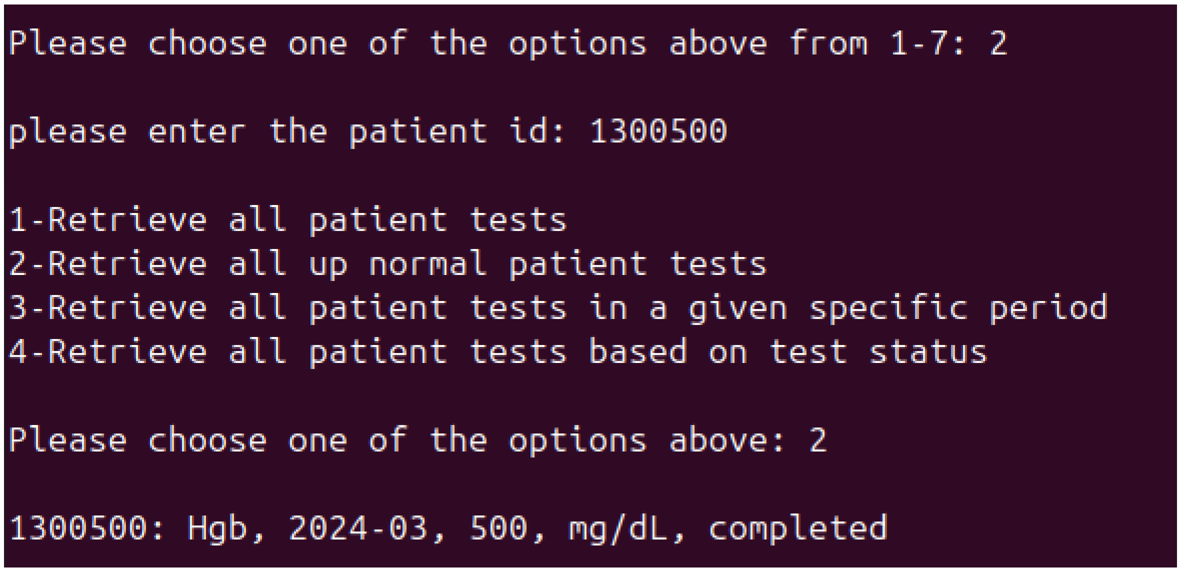
```
1300500: Hgb, 2024-07, 13.5, g/dL, completed
```

Note : we have inserted another rows to test the cases:


medicalRecord.txt

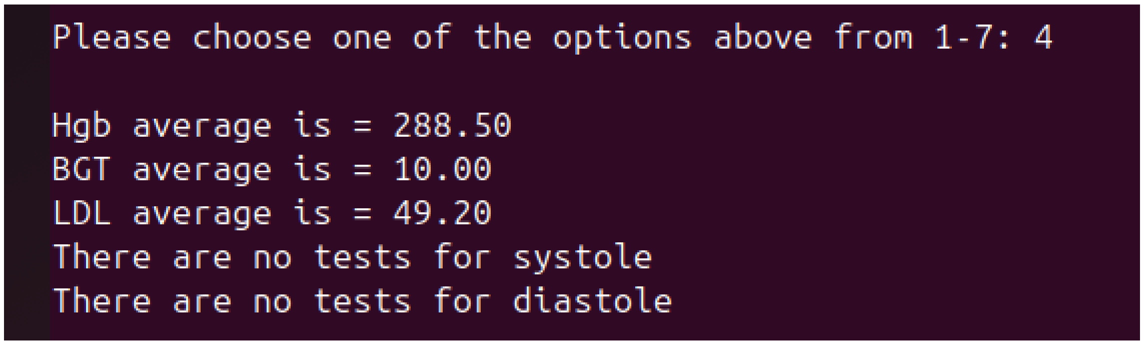
```
1300500: Hgb, 2024-03, 500, mg/dL, completed
1300500: LDL, 2024-07, 13.5, mg/dL, pending
1300550: BGT, 2024-07, 10, mg/dL, pending
1300700: LDL, 2024-07, 1.32, mg/dL, pending
1300450: LDL, 2024-07, 142, mg/dL, pending
1300545: LDL, 2024-07, 40, mg/dL, pending
1300771: Hgb, 2024-03, 77, mg/dL, completed
```

- Retrieving the up normal tests for 1300500 patient



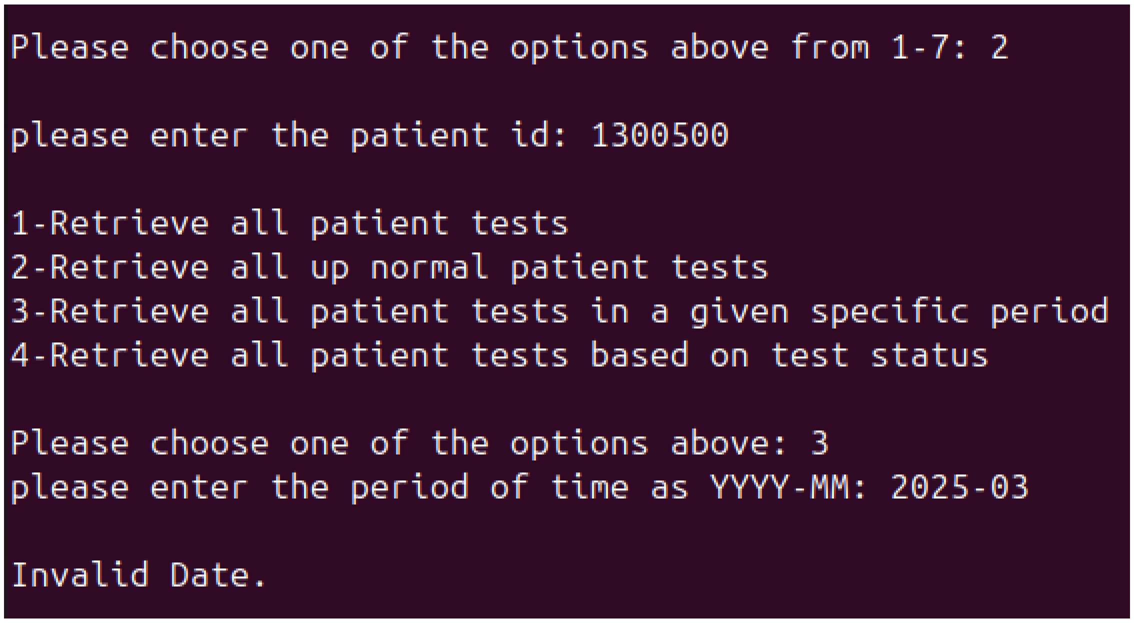
```
Please choose one of the options above from 1-7: 2

please enter the patient id: 1300500

1-Retrieve all patient tests
2-Retrieve all up normal patient tests
3-Retrieve all patient tests in a given specific period
4-Retrieve all patient tests based on test status

Please choose one of the options above: 2

1300500: Hgb, 2024-03, 500, mg/dL, completed
```

- Counting the average of the medical tests



```
Please choose one of the options above from 1-7: 4

Hgb average is = 288.50
BGT average is = 10.00
LDL average is = 49.20
There are no tests for systole
There are no tests for diastole
```

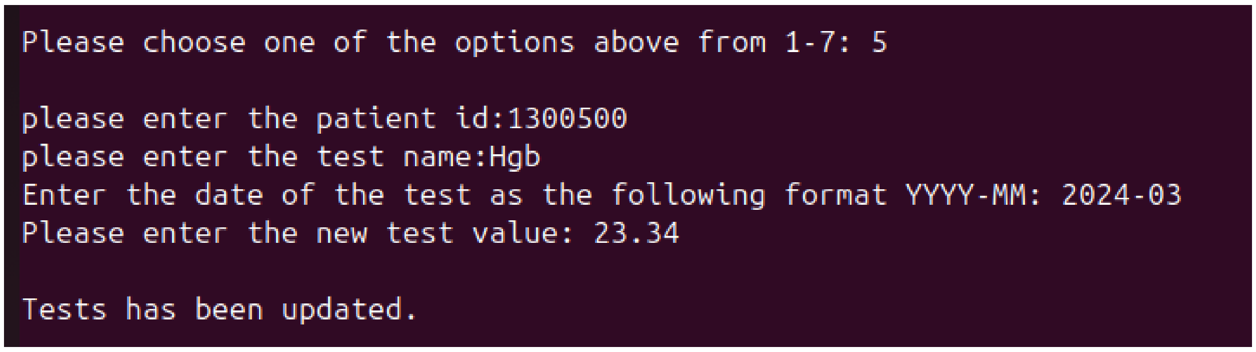
- Handling the invalid date inputs



```
Please choose one of the options above from 1-7: 2

please enter the patient id: 1300500

1-Retrieve all patient tests
2-Retrieve all up normal patient tests
3-Retrieve all patient tests in a given specific period
4-Retrieve all patient tests based on test status

Please choose one of the options above: 3
please enter the period of time as YYYY-MM: 2025-03

Invalid Date.
```

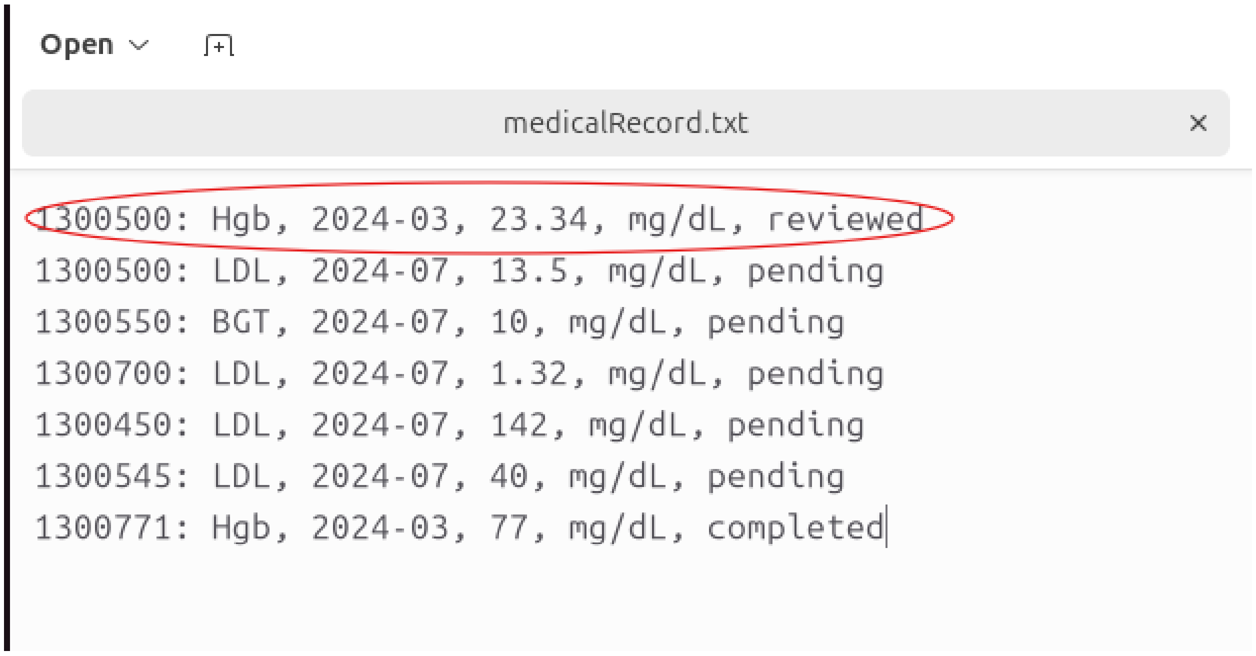
- Updating the Hgb test result for patient 1300500

Note that it was 500



```
Please choose one of the options above from 1-7: 5

please enter the patient id:1300500
please enter the test name:Hgb
Enter the date of the test as the following format YYYY-MM: 2024-03
Please enter the new test value: 23.34

Tests has been updated.
```

The file after updating:



```
1300500: Hgb, 2024-03, 23.34, mg/dL, reviewed
1300500: LDL, 2024-07, 13.5, mg/dL, pending
1300550: BGT, 2024-07, 10, mg/dL, pending
1300700: LDL, 2024-07, 1.32, mg/dL, pending
1300450: LDL, 2024-07, 142, mg/dL, pending
1300545: LDL, 2024-07, 40, mg/dL, pending
1300771: Hgb, 2024-03, 77, mg/dL, completed
```