**PROJECT1 REPORT**

|  |  |  |
| --- | --- | --- |
| 150117042 | 150117061 | 150117510 |
| Ezgi Doğruer | İsra Nur Alperen | Elif Gökpınar |

**PROGRAM 1**

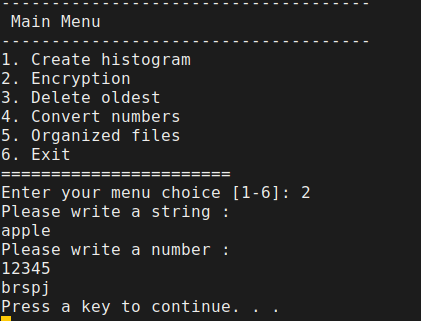
**STEPS :**

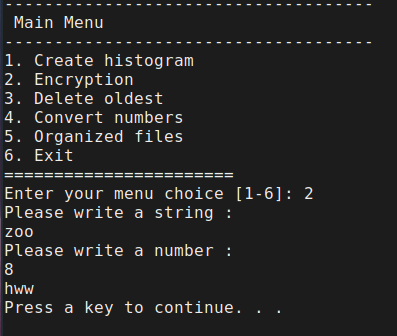
* Input file is read.Digits in the input file are transferred into array.
* ERROR CHECK STEP : If the numbers are negative or greater than 9, the program will exit.If there are no inputs, the program will also exit.
* New array are generated for controlling the input digits.The array contains valid numbers : (0,1,2….,9)
* In the end there are nested for loops to control if condition.If the valid number matches the input number , it puts the star.

|  |  |
| --- | --- |
| **EXAMPLE INPUT** | **OUTPUT** |
| **Screenshot from 2020-11-12 22-11-46.png** | **Screenshot from 2020-11-12 22-09-59.png** |

**PROGRAM 2**

**STEPS :**

* Program reads given inputs.
* First of all, I have defined arrays. One of them to putting in the letters of the word received, another array fort he number received and the last one for the alphabet.
* I filled in arrays using for to contain the characters of the word and number received.
* Then the length of the number I got could have been 1 or the length of the received word I used an if-else construct to check this.
* I compared the word array with the alphabet array index by index. I used the if statement to see if they encounter the same letter.
* For each letter in the string, I had to find another letter in the English alphabet advancing over the alphabet corresponding digit times.
* For this, I collected the index of the common letter with the added number and checked the return to the beginning in the alphabet. In line with this process, I subtracted the length of the alphabet from the total if it returns to the beginning of the alphabet.
* Finally, I put each new letter I found in a array and printed the new word array I obtained.



**PROGRAM 3**

**STEPS :**

* Firstly, we controled if there is a folder name as an input. If there is a folder name, the program open that folder. If no, program start to run on current directory.
* Secondly, we found the oldest file in the folder by using

oldest=$(find -type f -print0 | xargs -0 stat --format '%Y :%y %n' | sort | cut -d" " -f5-| head -1 | cut -d"/" -f2-) code.

* Lastly, we asked if you want to delete this file or not. If yes, we deleted it.

echo "Do you want to delete $oldest ? (y/n)"

read answer

if [ "$answer" = "y" ] || [ "$answer" = "Y" ];

then

rm $oldest

echo "$oldest is deleted"

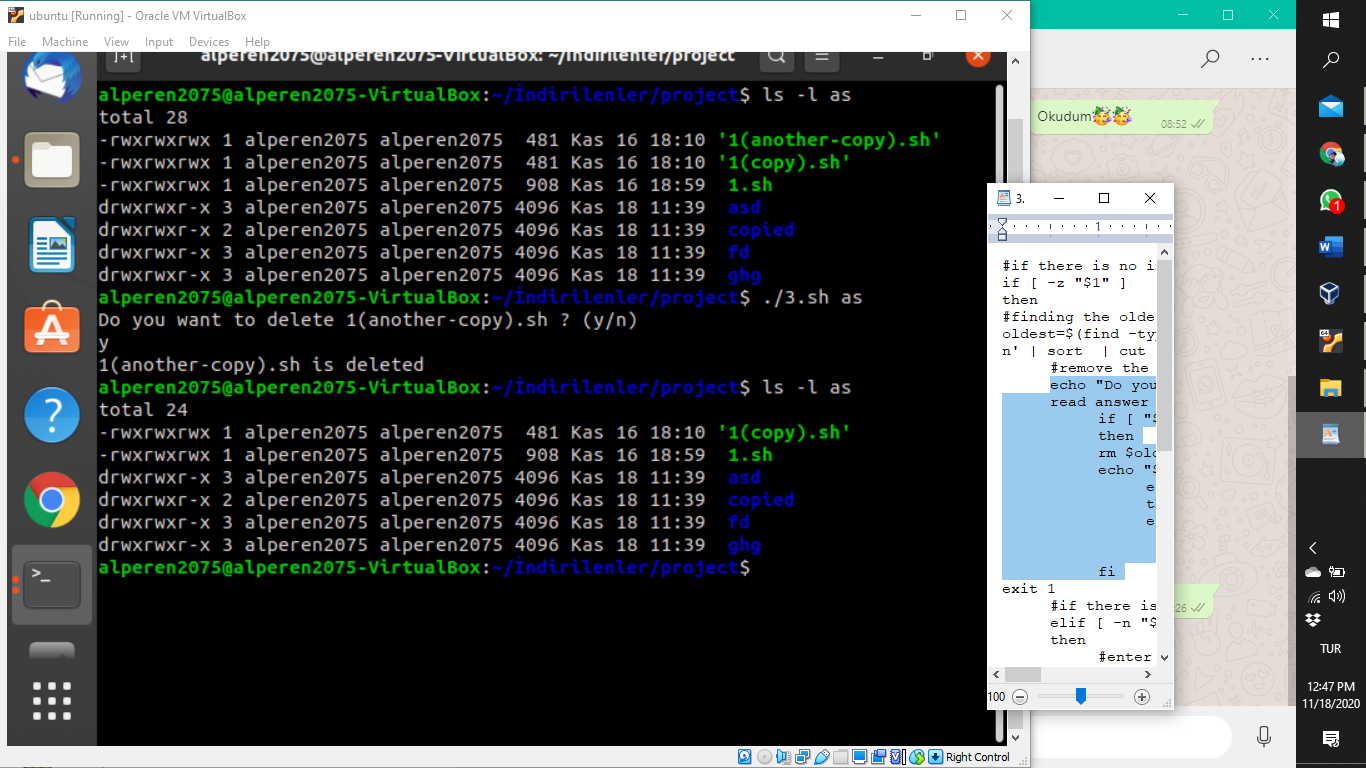
elif [ $answer = "n" ] || [ $answer = "N" ];

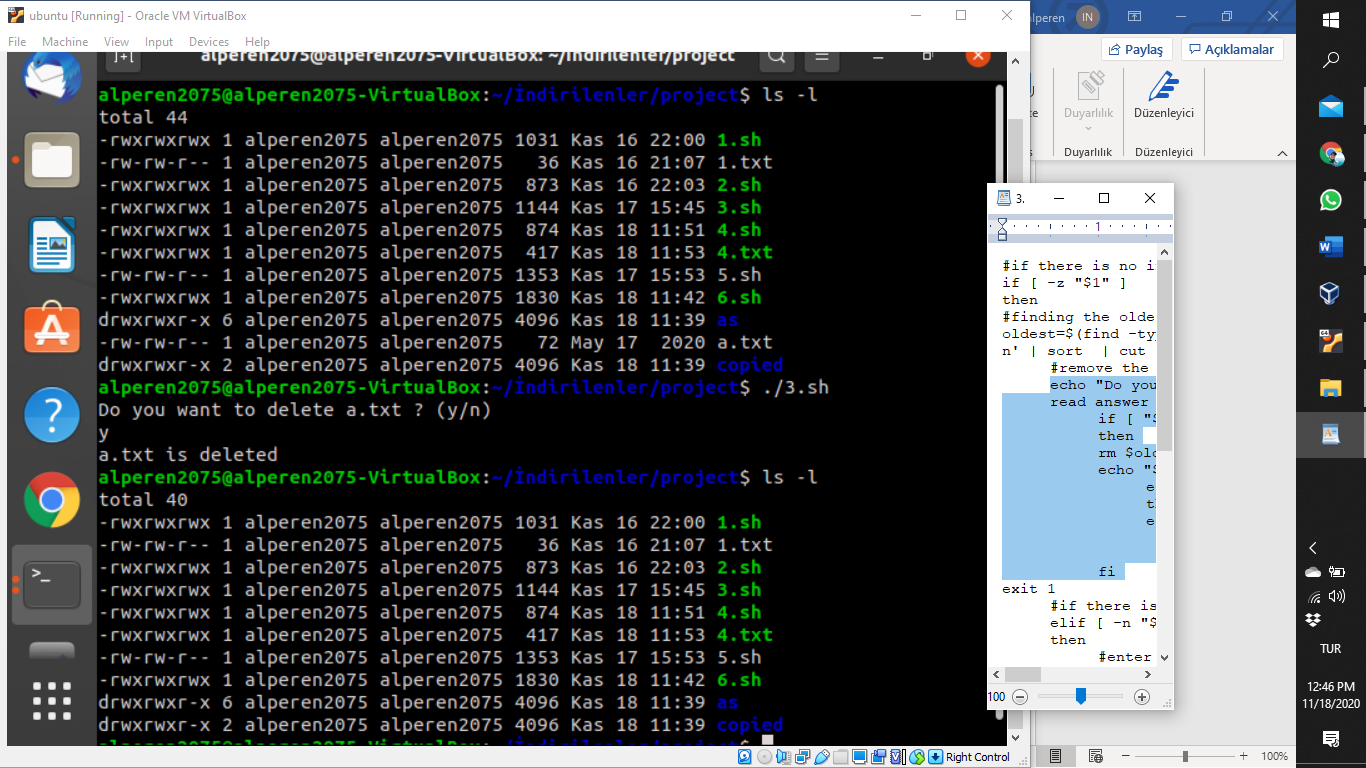
then

echo "$oldest is not deleted"

else

echo "wrong input"

 fi



**PROGRAM 4**

**STEPS :**

* Firstly, We read each word in the file. At the same time, we controlled words if include number. If a word include number, we send it in if statement.

re='^[0-9]+$'

#if words are number

if [[ $word =~ $re ]] ; then

last=""

for (( j=0 ; j<${#word} ; j++)) {

digit=${word:j:1}

case $digit in

0) num='zero' ;;

1) num='one' ;;

2) num='two' ;;

3) num='three' ;;

4) num='four' ;;

5) num='five' ;;

6) num='six' ;;

7) num='seven' ;;

8) num='eight' ;;

9) num='nine' ;;

esac

last="$last$num"

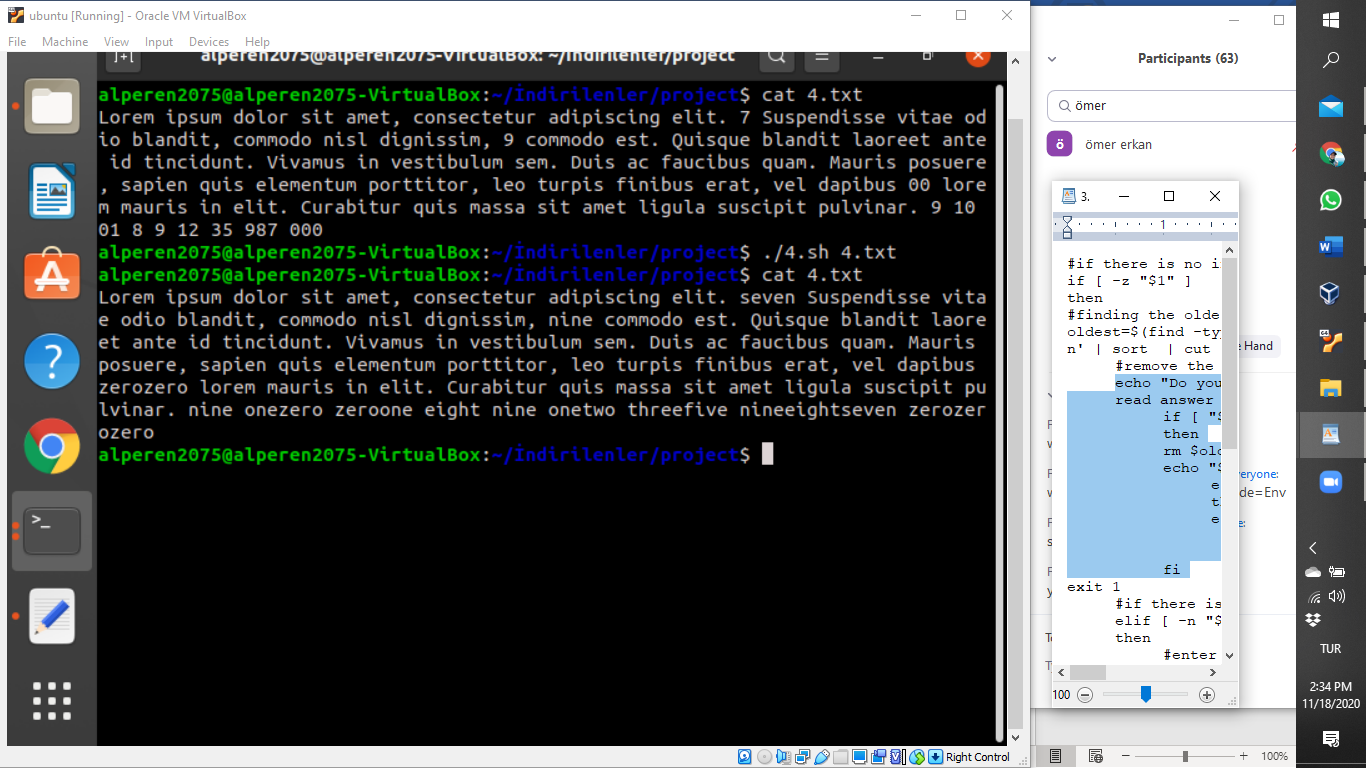
}

word="$last"

* In if statement, we created a loop to return for each number in word. And we assigned the value to num variable in case statement.
* Lastly, we added all word in an array. After that we writed array on the txt file.

#writing to file

> $file

echo ${arr[@]} >> $file

**PROGRAM 5**

**STEPS :**

* Program reads given input/inputs.
* Firstly, if there are “copied” directories from previous running of the code, the program deletes “copied” directories in order to prevent errors.The program uses “ rm –rf ”
* If there are 2 inputs (-R option and the input which is used wildcard property), program copies the files to ”copied” directory recursively.The program generates “copied” directory in each directory and their subdirectories.To generate new copied directory, the program uses “mkdir”.According to given input, we can copy some files.The user can decide this using wildcard.
* If there 1 inputs (the input which is used wildcard property), Copy operations are only done in current directory.

|  |
| --- |
| **$ ls** car.txt     file.txt    myprog5.sh    subdirectory csales.txt  myprog1.sh  practice.txt **$ ./myprog5.sh -R "c\*.txt"** |
| **$ ls** car.txt  csales.txt  myprog1.sh  practice.txt copied   file.txt    myprog5.sh  subdirectory **$ ls copied** car.txt  csales.txt **$ ls subdirectory** cent.txt  ceremony.txt  copied  exercise.txt  other **$ ls subdirectory/copied** cent.txt  ceremony.txt **$ ls subdirectory/other** copied  cstar.txt  fellow.txt **$ ls subdirectory/other/copied** cstar.txt |

**PROGRAM BONUS**

* Firstly, we created a menu.

1. Create histogram

2. Encryption

3. Delete oldest

4. Convert numbers

5. Organized files

6. Exit

* After that we created a switch case. And we took the inputs that programs wanted.
* Lastly, we implemented other programs to this program by using bash like;

2) echo "Please write a string :"

read input2

echo "Please write a number :"

read input3

bash ./2.sh $input2 $input3

echo "Press a key to continue. . ."; read ;;

