

# ԼԱԲՈՐԱՏՈՐ ԱՇԽԱՏԱՆՔ 1

**1. Ստեղծել lab1 անունով դիրեկտորիա, որում ստեղծել 4 տեքստային ֆայլ՝ file.txt, file1.txt, file2.txt, file3.txt և 2 դիրեկտորիա՝ dir1, dir2:**

```
mkdir lab1
cd lab1
touch file.txt file1.txt file2.txt file3.txt
mkdir dir1 dir2
```

**2. Կատարել հետևյալ հրամանները և բացատրել տարբերությունը.**

**ls \*.txt**

**ls \*.txt**

ls \*.txt # որոնում է \*.txt ֆայլերը  
ls \*.txt # որոնում է ինչ-որ .txt ֆայլ

**3. Ցուցադրել բոլոր տեքստային ֆայլերը (ավարտվում է .txt սիմվոլներով), որոնց անվանումը սկսվում է f սիմվոլով և բաղկացած է 4 սիմվոլից:**

```
ls f???*.txt
```

**4. Ցուցադրել բոլոր տեքստային ֆայլերը, որոնց անվանումը սկսվում է file բառով, որին հաջորդում է թվանշան:**

```
ls file[0-9]*.txt
```

**5. Ցուցադրել բոլոր տեքստային ֆայլերը, որոնց անվան վերջին սիմվոլը փոքրատառ է, կամ 1, 2 թվանշաններից որևէ մեկը:**

```
ls *[a-z0-1].txt
```

**6. Պատճենել /etc/passwd ֆայլը lab1 դիրեկտորիայում:**

```
cp /etc/passwd lab1/
```

**7. Անվանափոխել պատճենված ֆայլը՝ այն դարձնելով new: Տեղափոխել new ֆայլը դեպի dir1, այնուհետև տեղափոխել դեպի dir2:**

```
mv lab1/passwd lab1/new
mv lab1/new lab1/dir1/
mv lab1/dir1/new lab1/dir2/
```

**8. Սևիանափոխել *dir2*-ը՝ այն դարձնելով *dir3*: Տեղափոխել *dir3*-ը դեպի *dir1*:**

```
mv ./dir2 ./dir3  
mv ./dir3 ./dir1/
```

**9. Տեղափոխել *dir3*-ում գտնվող *new* ֆայլը դեպի *lab1*:**

```
mv ./dir1/new ./
```

**10. Ջնջել *lab1* դիրեկտորիան:**

```
rm -r lab1
```

## ԼԱԲՈՐԱՏՈՐ ԱՇԽԱՏԱՆՔ 2

*1. Կատարել հետևյալ հրամանները, բացատրել տարբերությունը.*

*cat > file1*

*cat file1*

*cat < file1*

cat > file1 # Ստեղծում է ֆայլ file1. և սպասում է ստեղնաշարի մուտքագրմանը մինչև Ctrl+D կմուտքագրվի: Մուտքագրված տեքստը կգրվի file1-ում:

cat file1 # Ցուցադրում է file1-ի բովանդակությունը:

cat < file1 # Ցուցադրում է file1-ի բովանդակությունը:

*2. Ստեղծել որոշակի պարունակությամբ 3 տեքստային ֆայլ՝ file1.txt, file2.txt, file3.txt: Այս ֆայլերի պարունակությունը կցել միմյանց և գրել նոր ֆայլում՝ final.txt:*

cat > file1.txt

cat > file2.txt

cat > file3.txt

cat file1.txt file2.txt file3.txt > Final.txt

*3. Ցուցադրել /home/student դիրեկտորիայում տեքստային ֆայլերի քանակը:*

ls /\*.txt | wc -l

*4. Սորտավորել file1.txt, file2.txt, file3.txt ֆայլերի պարունակությունը և գրել նոր ֆայլում:*

cat file1.txt file2.txt file3.txt | sort > sorted.txt

*5. Արտածել /home/student դիրեկտորիայի առաջին 15 ֆայլերը՝ դասավորված նվազման կարգով ըստ ֆայլի ծավալի:*

ls -S ./ | head -n 15

*Արտածել /home/student դիրեկտորիայի ֆայլերի ցուցակը, միաժամանակ այն տեղադրելով ls.txt ֆայլում:*

ls ./ > ls.txt

*7. Օգտագործելով tr հրամանը՝ զադտնագրել "secret text" տողը ROT13 ալգորիթմով: Վերծանել զադտնագրված հաղորդագրությունը:*

echo "secret text" | tr 'a-zA-Z' 'n-za-mN-ZA-M'

echo "frperg grkg" | tr 'a-zA-Z' 'n-za-mN-ZA-M'

**8. Ստեղծել նոր ֆայլ՝ *home.txt*, հետևյալ պարունակությամբ.**

***Student's home directory is {home\_dir}.***

***sed հրամանի միջոցով փոխարինել {home\_dir} հասցեով /home/student-նվ:***

```
echo "Student's home directory is {home_dir}." > home.txt
```

```
sed 's/{home_dir}/\home\student/' home.txt > home_modified.txt
```

**9. Ստեղծել ֆայլ, որն ունի առնվազն 5 տող պարունակություն՝ *file4.txt*: Արտածել 2-4-րդ տողերը:**

```
echo "Line 1" > file4.txt
```

```
echo "Line 2" >> file4.txt
```

```
echo "Line 3" >> file4.txt
```

```
echo "Line 4" >> file4.txt
```

```
echo "Line 5" >> file4.txt
```

```
sed -n '2p;4p' file4.txt
```

**10. Ջնջել *file4.txt* ֆայլի 2-4-րդ տողերը:**

```
sed -i '2d;4d' file4_modified.txt
```

# ԼԱԲՈՐԱՏՈՐ ԱՇԽԱՏԱՆՔ 3

**1. Յուզադրել /etc դիրեկտորիայի բոլոր ֆայլերը, որոնց անվան մեջ թվանշան կա:**

```
ls /etc | grep '[0-9]'
```

**2. Ստեղծել որոշակի պարունակությամբ ֆայլ: Ֆայլում փնտրել այլ տողերը, որոնք բաղակազմած են 5 տառից, և ավարտվում են 10 տառերով:**

```
echo -e "Hello\nWorld\nTestlo\nHillo\nOtherlo\n" >test_file.txt
grep -E '\b\w{3}lo$' test_file.txt
```

**3. Յուզադրել հետևյալ հրամանների տարբերությունը.**

```
grep -h '[A-Z]' dirlist*.txt
grep -h '[-AZ]' dirlist*.txt
grep -h '^[A-Z]' dirlist*.txt
grep -h '/^A-Z/' dirlist*.txt
```

grep -h '[A-Z]' список\_каталогов\*.txt: # Որոնում է առնվազն մեկ մեծատառ պարունակող տողեր:  
grep -h '[-AZ]' список\_каталогов\*.txt: # Որոնում է «-», «A» կամ «Z» նիշերից որևէ մեկը պարունակող տողեր:  
grep -h '^[A-Z]' список\_каталогов\*.txt: # Որոնում է մեծատառով սկսվող տողեր:  
grep -h '/^A-Z/' список\_каталогов\*.txt: # Որոնում է տողեր, որոնք պարունակում են մեծատառերից բացի որևէ այլ նիշ:

**4. dirlist-bin.txt և dirlist-sbin.txt ֆայլերում փնտրել այն տողերը, որոնք սկսվում են bz կամ zip բառերով:**

```
grep -E '^(bz|zip)' dirlist-bin.txt dirlist-sbin.txt
```

**5. Յուզադրել հետևյալ հրամանների տարբերությունը.**

```
grep -Eh '^(bz/gz/zip)' dirlist*.txt
grep -Eh '^bz/gz/zip' dirlist*.txt
```

grep -Eh '^(bz/gz/zip)' список\_каталогов\*.txt # Փնտրում է «bz», «gz» կամ «zip» տողեր:  
grep -Eh '^bz/gz/zip' список\_каталогов\*.txt # Որոնում է «bz»-ով սկսվող կամ «gz» կամ «zip» պարունակող տողեր:

**6. Ստուգել տրված էլ. հասցեի վավերականությունը: Օր. ' [test@test.com](mailto:test@test.com)**

```
echo "test@test.com" | grep -E '^([a-zA-Z0-9._%+-]+)@([a-zA-Z0-9.-]+\.[a-zA-Z]{2,})$'
#endregion
```

*7. Ստուգել տրված IPv4 հասցեի վավերականությունը: Օր.՝ 192.168.0.1*

```
echo "192.168.0.1" | grep -E '^([0-9]{1,3}\.){3}[0-9]{1,3}$'
```

*9. Տրված է հետևյալ ամսաթիվը. 01/02/1970, Ձևափոխել այն՝ բերելով հետևյալ ֆորմատի. 1970-02-01*

```
echo "01/02/1970" | sed -E 's~([0-9]+)/([0-9]+)/([0-9]+)~\3-\2-\1~'
```

*10. Տրված է հետևյալ հեռախոսահամարը. Օxx 12-34-56  
Ձևափոխել այն՝ բերելով հետևյալ ֆորմատի. (0xx) 123-456*

```
echo "0xx 12-34-56" | sed -E 's~^(0xx) ([0-9]{2})-([0-9])([0-9])-([0-9]{2})$~(\1) \2\3-\4\5~'
```

## ԼԱԲՈՐԱՏՈՐ ԱՇԽԱՏԱՆՔ 4

1. Ստեղծել նոր ֆայլ՝ *file.txt*, և դրա համար սահմանել հետևյալ իրավունքները.

- *rwX r-- r---*

```
touch file.txt
```

```
chmod 744 file.txt
```

2. Փոփոխել *file.txt* ֆայլի իրավունքները՝ բոլոր օգտատերերի համար ավելացնելով կատարելու իրավունք: Օգտվել ֆայլի ռեժիմների սիմվոլիկ ներկայացումից:

```
chmod a+x file.txt
```

3. Փոփոխել *file.txt* ֆայլի իրավունքները՝ խմբի օգտատերերի և այլ օգտատերերի համար հեռացնելով կատարելու իրավունքը: Օգտվել սիմվոլիկ ներկայացումից:

```
chmod go-x file.txt
```

4. Փոփոխել *file.txt* ֆայլի իրավունքները՝ հեռացնելով բոլոր իրավունքները, և սահմանելով գրելու և կարդալու իրավունքներ միայն ֆայլի սեփականատիրոջ համար: Նույն գործողությունը կատարել 2 անգամ՝ օգտվելով ֆայլի ռեժիմների 8-ական և սիմվոլիկ ներկայացումներից:

```
chmod 600 file.txt
```

5. Ստեղծել նոր դիրեկտորիա՝ *dir*: Սահմանել հետևյալ իրավունքները. *d rw- rw- rw-: dir* դիրեկտորիայում ստեղծել նոր ֆայլ՝ *file.txt*: Բացատրել ստացված հաղորդագրությունը: Փոփոխել դիրեկտորիայի իրավունքներն այնպես, որ ֆայլը հաջողությամբ ստեղծվի:

```
mkdir dir
```

```
chmod 666 dir
```

```
touch dir/file.txt
```

6. *dir* դիրեկտորիայում ստեղծել ևս 2 ֆայլ՝ *file1.txt*, *file2.txt*: Մեկ հրամանի կատարմամբ փոփոխել դիրեկտորիայի բոլոր ֆայլերի իրավունքները՝ սահմանելով - *rwX rw- rw-*:

```
touch dir/file1.txt dir/file2.txt
```

```
chmod 766 dir/*
```

7. *umask* հրամանի միջոցով սահմանել այնպիսի *bitmask*, որ նոր ստեղծվող ֆայլերը լռելյայն կերպով ունենան հետևյալ իրավունքները.

- *rw- r-- ---*

umask 027

**8. umask հրամանի միջոցով սահմանել այնպիսի bitmask, որ նոր ստեղծվող ֆայլերը լռելյայն կերպով ունենան հետևյալ իրավունքները.**

**- rw- --- ---: Վերադարձնել bitmask-ի**

**լռելյայն արժեքը՝ կատարելով umask 0022 հրամանը:**

umask 077

umask 022

**9. Ստեղծել սկրիպտ, որը կարտածի /home/student դիրեկտորիայի ֆայլերի ցուցակը:**

**Սկրիպտի համար սահմանել - -wx r-- r-- իրավունքները: Կատարել սկրիպտը և բացատրել ստացված հաղորդագրությունը: Փոփոխել իրավունքներն այնպես, որ սկրիպտը կատարվի:**

```
echo 'ls /home/student' > script.sh
```

```
chmod 741 script.sh
```

```
./script.sh
```

```
chmod +x script.sh
```

**10. Սկրիպտը տեղադրել ~/bin դիրեկտորիայում: Փոփոխել \$PATH փոփոխականն այնպես, որ սկրիպտը հնարավոր լինի կատարել առանց հասցեն նշելու:**

```
mkdir ~/bin
```

```
mv script.sh ~/bin
```

```
echo 'export PATH=$PATH:~/bin' >> ~/.bashrc
```

```
source ~/.bashrc
```



## ԼԱԲՈՐԱՏՈՐ ԱՇԽԱՏԱՆՔ 5

1. Ստեղծել կամայական արժեքով փոփոխական `str`: Ցուցադրել հետևյալ հրամանների աշխատանքի արդյունքի տարբերությունը.

```
echo $str2
echo ${str}2
```

```
str="Hello"
echo str2 #str2
echo ${str}2 #Hello2
```

2. Արտածել հետևյալ արտահայտության արժեքը.

```
52*4
10
```

```
echo $(5 ** 2 * 4 / 10)
```

3. Ցուցադրել հետևյալ հրամանների աշխատանքի արդյունքի տարբերությունը.

```
echo "The balance for user $USER is: $5.00"
echo "The balance for user $USER is: \ $5.00"
```

```
echo "The balance for user $USER is: $5,00" # The balance for user root is: ,00
echo "The balance for user $USER: \ $5,00" # The balance for user root: $5,00
```

4. Ցուցադրել հետևյալ հրամանների աշխատանքի արդյունքի տարբերությունը.

```
echo '$USER $((2 + 2)) $(ls)'
echo "$USER $((2 + 2)) $(ls)"
```

```
echo '$USER $((2 + 2)) $(ls)' # $USER $((2 + 2)) $(ls)
echo "$USER $((2 + 2)) $(ls)" # root 4 labs main.sh
```

5. Առանձին տողերում արտածել ընթացիկ դիրեկտորիան և այնտեղ առկա տեքստային ֆայլերի քանակը: Օգտագործել *Here document*:

```
current_dir=$(pwd)
num_files=$(ls -l | grep -c "^-")
cat <<EOF
Current directory: $current_dir
Number of files: $num_files
EOF
```

6. Գրել 2 թվեր գումարող ֆունկցիա: Թվերը ֆունկցիային փոխանցել որպես

***արգումենտներ:***

```
sum_numbers() {  
    result=$(( $1 + $2 ))  
    echo "The summary $1 and $2 is equal to $result"  
}  
sum_numbers 5 7
```

***7. Գրել ֆունկցիա, որը որպես արգումենտ կստանա դիրեկտորիայի անվանումը և կարտածի այնտեղ առկա ֆայլերի քանակը:***

```
count_files() {  
    dir_name=$1  
    num_files=$(ls -l $dir_name | grep -c "^-")  
    echo "Number of files in the directory $dir_name: $num_files"  
}
```

```
count_files ./
```

***8. Գրել սկրիպտ, որում նոր ստեղծված ֆայլի համար ստուգել, թե արդյոք այն ունի կարդալու / գրելու / կատարելու թույլտվություն, և արտածել համապատասխան հաղորդագրություն:***

```
touch new_file.txt  
if [ -e new_file.txt ]; then  
    chmod +rwx new_file.txt  
    echo "File new_file.txt has been created, and permissions for read, write, and execute are set."  
else  
    echo "Error: File new_file.txt was not created."  
fi
```

***9. Գրել սկրիպտ, որը կորոշի տրված 2 թվերից մեծագույնը և կարտածի:***

```
find_max() {  
    if [ $1 -gt $2 ]; then  
        echo "The largest number is: $1"  
    else  
        echo "The largest number is: $2"  
    fi  
}
```

```
find_max 12 8
```

***10. Գրել սկրիպտ, որը կորոշի տրված թիվը զույգ է, թե կենտ: Արտածել համապատասխան հաղորդագրություն:***

```
check_even_odd() {  
    if [ $((($1 % 2)) -eq 0 ]; then  
        echo "$1 is an even number."  
    else  
        echo "$1 is an odd number."  
    fi  
}
```

```
check_even_odd 7
```

## ԼԱԲՈՐԱՏՈՐ ԱՇԽԱՏԱՆՔ 6

**1. Գրել սկրիպտ, որը տրված թվի համար կարտածի հաղորդագրություն այն մասին, արդյոք թիվը դրական է, բացասական, թե՛ 0: Թիվը վավերացնել ռեզուլյար արտահայտության միջոցով: Պայմանների ստուգման համար օգտագործել [[ ]]**  
**օպերատորը:**

```
read -r -p "Enter a number: " number
```

```
if [[ $number -gt 0 ]]; then
    echo "The number is positive."
elif [[ $number -lt 0 ]]; then
    echo "The number is negative."
else
    echo "The number is zero."
Fi
```

**2. Կատարել 1-ին կետը՝ պայմանների ստուգման համար օգտագործելով (( ))**  
**օպերատորը:**

```
read -r -p "Enter a number: " number
```

```
if ((number > 0)); then
    echo "The number is positive."
elif ((number < 0)); then
    echo "The number is negative."
else
    echo "The number is zero."
Fi
```

**3. Գրել սկրիպտ, որը կստուգի, թե արդյոք տրված թիվը 2-ի, 3-ի և 5-ի բազմապատիկ է:**

```
read -r -p "Enter a number: " number
```

```
if ((number % 2 == 0 && number % 3 == 0 && number % 5 == 0)); then
    echo "The number is divisible by 2, 3, and 5."
else
    echo "The number is not divisible by 2, 3, and 5."
Fi
```

**4. Գրել սկրիպտ, որը կստուգի, թե արդյոք տրված թիվը 2-ի, 3-ի կամ 5-ի բազմապատիկ է:**

```

read -r -p "Enter a number: " number

if ((number % 2 == 0 || number % 3 == 0 || number % 5 == 0)); then
    echo "The number is divisible by 2, 3, or 5."
else
    echo "The number is not divisible by 2, 3, or 5."
fi

```

**5. Մեկ հրամանի միջոցով ստեղծել նոր ֆայլ և այդ ֆայլին ավելացնել կատարելու թույլտվություն:**

```

touch new_file.txt
chmod +rwx new_file.txt

```

**6. Մեկ հրամանի միջոցով ստուգել, թե արդյոք գոյություն ունի dir անունով դիրեկտորիա, և, եթե գոյություն չունի, ապա ստեղծել:**

```

[ -d dir ] || mkdir dir

```

**7. Գրել սկրիպտ, որը կստուգի, թե արդյոք տրված թիվը գտնվում է սահմանված միջակայքում: Ստուգվող թիվը և միջակայքի սահմանների թվերը ներմուծել read հրամանի միջոցով: Իրականացնել տվյալների վավերացում:**

```

read -r -p "Enter a number: " number

if ((number >= 10 && number <= 20)); then
    echo "The number is within the range of 10 to 20."
else
    echo "The number is not within the range of 10 to 20."
fi

```

**8. Գրել սկրիպտ, որը որպես ներմուծվող արժեք կստանա մեկ բառ, և կստուգի արդյոք այն համընկնում է "Secret" բառի հետ: read հրամանը կատարել այնպես, որ ներմուծված արժեք պահպանվի REPLY փոփոխականի մեջ, իսկ ներմուծումն իրականացնելիս տառերը չցուցադրվեն էկրանին:**

```

read -r -p "Enter a word: " word

if [ "$word" == "Secret" ]; then
    echo "The word matches 'Secret'."
else
    echo "The word does not match 'Secret'."

```

fi

*9. Գրել սկրիպտ, որը որպես ներմուծվող արժեք կստանա ֆայլի անուն: Վավերացնել անունը հետևյալ կանոններով. կարող է պարունակել տառեր, թվեր,*

*-  
.*

*\_ սիմվոլները: Եթե նման ֆայլ գոյություն չունի, ապա ստեղծել: Ցուցադրել համապատասխան հաղորդագրություն ֆայլի ստեղծման կամ առկայության մասին:*

```
read -r -p "Enter a filename: " filename
```

```
if [[ "$filename" =~ ^[A-Za-z0-9._-]+$ ]]; then
```

```
    touch "$filename"
```

```
    echo "File $filename has been created."
```

```
else
```

```
    echo "Invalid filename."
```

```
fi
```

*10. Գրել <<Հաշվիչ>> ծրագիր, որը կստանա ճիշտ 3 արժեք հետևյալ հաջորդականությամբ. թիվ, գործողություն, թիվ: Օր՝ 2 + 3: Ստուգել ներմուծված արժեքների քանակը, և 3-ից տարբեր լինելու դեպքում արտածել հաղորդագրություն սխալի մասին: Երականացնել արժեքների վավերացում: Գործողության համար սահմանել հետևյալ ընդունելի արժեքները. + - \* / \*\*: Արտածել գործողության արդյունքը:*

```
read -r -p "Enter the first number: " num1
```

```
read -r -p "Enter the operator (+, -, *, /, **): " operator
```

```
read -r -p "Enter the second number: " num2
```

```
case $operator in
```

```
+) result=$((num1 + num2)) ;;
```

```
-) result=$((num1 - num2)) ;;
```

```
\*) result=$((num1 * num2)) ;;
```

```
/) result=$((num1 / num2)) ;;
```

```
\*\*) result=$((num1 ** num2)) ;;
```

```
*) echo "Invalid operator." ;;
```

```
esac
```

```
echo "Result of the operation: $result"
```

# ԼԱԲՈՐԱՏՈՐ ԱՇԽԱՏԱՆՔ 7

**1. Գրել ծրագիր, որը կարտածի [0, 20] միջակայքի զույգ թվերը: Օգտագործել while ցիկլ:**

```
count=0
while [ $count -le 20 ]; do
    echo $count
    count=$((count + 2))
done
```

**2. Կատարել 1-ին առաջադրանքը՝ օգտագործելով until ցիկլ:**

```
count=0
until [ $count -gt 20 ]; do
    echo $count
    count=$((count + 2))
done
```

**3. Գրել ծրագիր, որը գտնվում է անվերջ ցիկլի մեջ: Ցիկլի յուրաքանչյուր իտերացիայում արտածել տվյալ ինդեքսի քառակուսին: Ցիկլը ավարտել այն պահին, երբ արտածվող թիվը կգերազանցի 1000: Օգտագործել while ցիկլ:**

```
index=0
while true; do
    square=$((index * index))
    if [ $square -gt 1000 ]; then
        break
    fi
    echo "Square of $index is $square"
    index=$((index + 1))
done
```

**4. Կատարել 1-ին առաջադրանքը՝ օգտագործելով for ցիկլ:**

```
for ((count = 0; count <= 20; count += 2)); do
    echo $count
done
```

**5. Գրել ծրագիր, որը որպես մուտքային տվյալ կստանա ամբողջ թիվ և կարտածի դրա կրկնապատիկը: Ծրագիրը սկսելիս ցուցադրել հետևյալ հաղորդագրությունը .**

**Enter an integer number, or enter q to quit:**

Ամբողջ թիվ մուտքագրելուց հետո ցուցադրել դրա կրկնապատիկը, և կրկին ցուցադրել

հաղորդագրությունը՝ մինչև q տառի սեղմումը:

```
while true; do
    read -r -p "Enter an integer or 'q' to exit: " input
    if [ "$input" == "q" ]; then
        break
    fi
    echo "Double of $input is $((input * 2))"
done
```

**6. Գրել ծրագիր, որը կստեղծի /home/student դիրեկտորիայի տեքստային ֆայլերի անունները պարունակող ֆայլ: while ցիկլի միջոցով համարակալել այս ֆայլերի անունները և արտածել:**

```
index=1
while [ $index -le 5 ]; do
    touch ./file_$index.txt
    index=$((index + 1))
done
```

**7. Գրել ծրագիր, որը for ցիկլի միջոցով կարտածի [0, 30] միջակայքում գտնվող 3-ի բազմապատիկ թվերը:**

```
for ((i = 1; i <= 3; i++)); do
    echo $((RANDOM % 31))
done
```

**8. Գրել ծրագիր, որը for ցիկլի միջոցով կարտածի ընթացիկ դիրեկտորիայի բոլոր տեքստային ֆայլերի անունները:**

```
for file in *; do
    echo "$file"
done
```

**9. Գրել ծրագիր, որը որպես արգումենտ կստանա ֆայլերի անուններ, և այդ ֆայլերից յուրաքանչյուրի համար կարտածի ֆայլի ամենակարճ բառը և դրա երկարությունը:**

```
for file in "$@"; do
    shortest_word=$(awk '{ print length, $0 }' "$file" | sort -n | head -n 1 | cut -d" " -f2-)
    echo "Shortest word in $file: $shortest_word"
done
```



*10. Գրել ծրագիր, որը որպես արգումենտներ կստանա ֆայլերի անուններ, և դրանցից  
յուրաքանչյուրի համար կստուգի, թե արդյո՞ք տվյալ ֆայլը գոյություն ունի, թե ոչ:  
Գոյություն ունենալու դեպքում ստուգել, թե արդյո՞ք ֆայլն ունի կարդալու  
թույլտվություն: Արտածել համապատասխան հաղորդագրություններ:*

```
for file in "$@"; do
    if [ -e "$file" ]; then
        echo "$file exists."
        if [ -r "$file" ]; then
            echo "$file is readable."
        else
            echo "$file is not readable."
        fi
    else
        echo "$file does not exist."
    fi
done
```

## ԼԱԲՈՐԱՏՈՐ ԱՇԽԱՏԱՆՔ 7

1. Գրել ծրագիր, որը ինտերակտիվ ռեժիմում կստանա [0, 3] միջակայքի որևէ թիվ, և արգումենտի արժեքից կախված կկատարի հետևյալ գործողությունները.

□ 0 – ավարտել ծրագիրը:

□ 1 – արտածել /home/student դիրեկտորիայի ֆայլերի ցուցակը:

□ 2 – արտածել /home/student դիրեկտորիայի տեքստային ֆայլերի քանակը:

□ 3 – արտածել աշխատանքային դիրեկտորիան: Գործողությունները կրկնել մինչև 0 արժեքի ներմուծումը: Օգտագործել case օպերատոր:

```
while true; do
    echo "Enter a number (0 to exit): "
    read -r num

    case $num in
        0)
            echo "Program terminated."
            break
            ;;
        1)
            cp * ./
            echo "List of files copied to ./."
            ;;
        2)
            count=$(ls /*.txt | wc -l)
            echo "Number of text files in .: $count."
            ;;
        3)
            pwd
            ;;
        *)
            echo "Invalid number. Please enter 0, 1, 2, or 3."
            ;;
    esac
done
```

**2. Գրել ծրագիր, որը որպես արգումենտ կստանա տարվա ամսի անունը և կվերադարձնի տվյալ ամսում առկա օրերի քանակը: Օգտագործել case օպերատոր:**

```
echo "Enter the month name: "
read -r month

case $month in
    "January" | "March" | "May" | "July" | "August" | "October" | "December")
        echo "The month $month has 31 days."
        ;;
    "April" | "June" | "September" | "November")
        echo "The month $month has 30 days."
        ;;
    "February")
        echo "The month $month has 28 or 29 days depending on whether it's a leap year."
        ;;
    *)
        echo "Invalid month name."
        ;;
Esac
```

**3. Գրել <<Հաշվիչ>> ծրագիր, որը կստանա ճիշտ 3 արժեք հետևյալ հաջորդականությամբ. թիվ, գործողություն, թիվ: Օր՝ 2 + 3: Արժեքները ներմուծել ինտերակտիվ ռեժիմում (read հրամանի միջոցով): Գործողության համար սահմանել հետևյալ ընդունելի արժեքները. + - \*/ \*\*: Արտածել գործողության արդյունքը: Գործողության տիպը որոշելու համար օգտագործել case օպերատոր:**

```
while true; do
    echo "Enter the first number: "
    read -r num1

    echo "Enter the operation (+, -, *, /, **): "
    read -r operation

    echo "Enter the second number: "
    read -r num2

    case $operation in
        "+")
            result=$((num1 + num2))
            ;;
        "-")
            result=$((num1 - num2))
            ;;
        "**")
            result=$((num1 ** num2))
            ;;
    esac
    echo "Result: $result"
done
```

```

        result=$((num1 * num2))
        ;;
    "/" )
        result=$((num1 / num2))
        ;;
    "**" )
        result=$((num1 ** num2))
        ;;
    *)
        echo "Invalid operation."
        continue
        ;;
esac

echo "Result: $result"
done

```

**4. Գրել ծրագիր, որը որպես հրամանային տողի արգումենտ կստանա ֆայլի անուն: Եթե ֆայլը գոյություն ունի և ունի կարդալու թույլտվություն, ապա արտածել ֆայլի պարունակությունը, հակառակ դեպքում արտածել համապատասխան հաղորդագրություն:**

```

if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <filename>"
    exit 1
fi

filename="$1"

if [ -r "$filename" ]; then
    cat "$filename"
else
    echo "File does not exist or is not readable."
fi

```

**5. Գրել ծրագրի պատճենը՝ առանց ծր հրամանից օգտվելու: Ծրագիրը պետք է որպես արգումենտներ ստանա 2 ֆայլերի անուններ, և պատճենի առաջին ֆայլի պարունակությունը 2-րդի մեջ: Իրականացնել ծրագրի պատշաճ աշխատանքի համար հարկավոր ստուգումներ (ֆայլը գոյություն ունի, ունի կարդալու թույլտվություն):**  
**echo "Enter the first filename: "**

```

read -r file1

echo "Enter the second filename: "
read -r file2

```

```
if [ -r "$file1" ]; then
    cat "$file1" >"$file2"
    echo "Content of the file successfully copied."
else
    echo "File $file1 does not exist or is not readable."
Fi
```

**7. Գրել ծրագիր, որը որպես հրամանային տողի արգումենտներ կստանա կամայական քանակով արժեքներ և նույնությամբ կարտածի այդ արժեքները: Օգտագործել while ցիկլ և shift օպերատոր:**

```
count=1

while [ "$#" -gt 0 ]; do
    echo "Argument $count: $1"
    count=$((count + 1))
    shift
done
```

**8.Իրականացնել 7-րդ կետը՝ օգտագործելով for ցիկլ և \$@ հատուկ փոփոխականը:**

```
for arg in "$@"; do
    echo "Argument: $arg"
done
```

**9. Գրել ծրագիր, որը որպես հրամանային տողի արգումենտներ կստանա կամայական քանակով թվեր և կարտածի այդ թվերի գումարը:**

```
sum=0

for arg in "$@"; do
    if [[ "$arg" =~ ^[0-9]+$ ]]; then
        sum=$((sum + arg))
    else
        echo "Invalid input. Please enter valid numbers."
        exit 1
    fi
done

echo "Sum: $sum"
```

**10. Գրել ծրագիր, որը որպես հրամանային տողի արգումենտներ կստանա կամայական**

*քանակով թվեր և կարտածի գույգ թվերի քանակը:*

even\_count=0