

```
#!/bin/bash

echo "Enter the value of n:"
read n

echo "The expected pattern is:"

for (( i=1; i<=n; i++ ))
do
    for (( j=1; j<=i; j++ ))
        do
            echo -n "$j "
        done
        echo ""
    done
done
```

```
shubham@DESKTOP-670QJ40:~$ cd shell
shubham@DESKTOP-670QJ40:~/shell$ vi q1.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q1.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q1.sh
Enter the value of n:
6
The expected pattern is:
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

```
#!/bin/bash

echo "Enter the numbers to addition:"

read num1 num2

sum=$(($num1 + $num2))

echo "The sum of $num1 and $num2 is $sum"
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q3.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q3.sh
Enter the numbers to addition:
5 5
The sum of 5 and 5 is 10
```

```

# !/bin/bash

echo "Enter Two numbers : "
read a
read b

echo "Enter Choice :"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
read ch

case $ch in
    1)res=$(( $a + $b ))
       ;;
    2)res=$(( $a - $b ))
       ;;
    3)res=$(( $a * $b ))
       ;;
    4)res=$(( $a / $b ))
       ;;
esac
echo "Result : $res"

```

```

shubham@DESKTOP-670QJ40:~/shell$ vi q4.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q4.sh
Enter Two numbers :
45
54
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
1
Result : 99
shubham@DESKTOP-670QJ40:~/shell$ vi q4.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q4.sh
Enter Two numbers :
10
2
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
4
Result : 5

```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q4.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q4.sh
Enter Two numbers :
5
2
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
3
Result : 10
shubham@DESKTOP-670QJ40:~/shell$ ./q4.sh
Enter Two numbers :
4
2
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
2
Result : 2
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q4.sh
shubham@DESKTOP-670QJ40:~/shell$ vi q5.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q5.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q5.sh
Usage: ./q5.sh arg1 arg2 ... argn
shubham@DESKTOP-670QJ40:~/shell$ vi q5.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q5.sh 10 20 30
The largest value is 30
```

```
#!/bin/bash

if [ $# -eq 0 ]; then
    echo "Usage: $0 arg1 arg2 ... argn"
    exit 1
fi

largest=$1

for arg in "$@"; do
    if [ "$arg" -gt "$largest" ]; then
        largest=$arg
    fi
done

echo "The largest value is $largest"
```

```
#!/bin/bash

if [ $# -eq 0 ]; then
    echo "Usage: $0 number"
    exit 1
fi

number=$1

reverse=""
while [ "$number" -gt 0 ]; do
    remainder=$(( $number % 10 ))
    reverse="$reverse$remainder"
    number=$(( $number / 10 ))
done

echo "The reversed number of entered number is $reverse"
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q6.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q6.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q6.sh 122
The reversed number of entered number is 221
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q2.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q2.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q2.sh
Enter the value of n:
5
The expected pattern is:
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

```
#!/bin/bash

echo "Enter the value of n:"
read n

echo "The expected pattern is:"


num=1
row=1

while [ $row -le $n ]
do

    for (( i=1; i<=row; i++ ))
        do
            echo -n "$num "
            num=$((num+1))
            done

            echo ""
            row=$((row+1))
            done
```

```
#!/bin/bash

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

if [ ! -f "$1" ]; then
    echo "Error: $1 is not a regular file or does not exist."
    exit 1
fi

sed -i '/^$\s*/d' "$1"

echo "All empty lines of the file $1 will be deleted"

echo "Before script running, content of the file $1 :"
cat "$1"

echo "After script running, content of the file $1 :"
cat "$1"
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q7.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q7.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q7.sh first.sh
All empty lines of the file first.sh will be deleted
Before script running, content of the file first.sh :
#!/bin/bash
echo hello world
After script running, content of the file first.sh :
#!/bin/bash
echo hello world
```

```

#!/bin/bash

if [ $# -eq 0 ]; then
    echo "Usage: $0 string"
    exit 1
fi

number=${1%[-*/]}
operator=${1:${#number}:1}

if [[ ! $operator =~ [-*/] ]]; then
    echo "Error: Invalid operator."
    exit 1
fi

result=0
for ((i=0; i<${#number}; i++)); do
    digit=${number:$i:1}
    case $operator in
        +) result=$((result + digit)) ;;
        -) result=$((result - digit)) ;;
        \*) result=$((result * digit)) ;;
        /) result=$((result / digit)) ;;
    esac
done

echo "The $operator of the digits in $number is $result"

```

```

shubham@DESKTOP-670QJ40:~/shell$ vi q8.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q8.sh 12345+
The + of the digits in 12345 is 15

```

```
#!/bin/bash

read -p "Enter limit for Fibonacci series: " n

a=0
b=1

echo -n "$a"

while [ $b -le $n ]; do
    echo -n ", $b"
    c=$((a + b))
    a=$b
    b=$c
done

echo ""
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q9.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q9.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q9.sh
Enter limit for Fibonacci series: 10
0, 1, 1, 2, 3, 5, 8
```

```

#!/bin/bash

read -p "Enter a string: " input_string

string_array=($input_string)

echo "The lengths of each string are as below:"
for word in "${string_array[@]}"; do
    echo "Length of the string($word) - ${#word}"
done

```

```

shubham@DESKTOP-670QJ40:~/shell$ vi q10.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q10.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q10.sh
Enter a string: shubham bal
The lengths of each string are as below:
Length of the string(shubham) -7
Length of the string(bal) -3

```

```

#!/bin/bash

rows=8
cols=8

for (( row=0; row<$rows; row++ )); do
    for (( col=0; col<$cols; col++ )); do
        if (( ($row+$col) % 2 == 0 )); then
            echo -n "1 "
        else
            echo -n "0 "
        fi
    done
    echo ""
done

```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q11.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q11.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q11.sh
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
```

```
#!/bin/bash

while getopts "a:d:" opt; do
    case ${opt} in
        a )
            order="ascending"
            ;;
        d )
            order="descending"
            ;;
        \? )
            echo "Invalid option: -$OPTARG" 1>&2
            exit 1
            ;;
        : )
            echo "Option -$OPTARG requires an argument." 1>&2
            exit 1
            ;;
    esac
done
shift $((OPTIND -1))

args=("$@")

if [[ "$order" == "ascending" ]]; then
    sorted_args=$(printf '%s\n' "${args[@]}" | sort -n)
elif [[ "$order" == "descending" ]]; then
    sorted_args=$(printf '%s\n' "${args[@]}" | sort -nr)
fi

echo "The $order order of the array is: "
echo "${sorted_args[@]}"
```

```

shubham@DESKTOP-670QJ40:~/shell$ vi q12.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q12.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q12.sh
The order of the array is:

shubham@DESKTOP-670QJ40:~/shell$ ./q12.sh 12 54 65 34 95 75 82
The order of the array is:

shubham@DESKTOP-670QJ40:~/shell$ vi q12.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q12.sh a 12 54 65 34 95 75 82
The order of the array is:

shubham@DESKTOP-670QJ40:~/shell$ vi q12.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q12.sh 12 54 65 34 95 75 82 a
The order of the array is:

shubham@DESKTOP-670QJ40:~/shell$ ./q12.sh -a 12 54 65 34 95 75 82
The ascending order of the array is:
34 54 65 75 82 95
shubham@DESKTOP-670QJ40:~/shell$ ./q12.sh -d 12 54 65 34 95 75 82
The descending order of the array is:
95 82 75 65 54 34

```

```

#!/bin/bash

# Function to rename a file/directory with lower case letters
rename_lowercase() {
    local old_name="$1"
    local new_name=$(echo "$old_name" | tr '[:upper:]' '[:lower:]')
    if [[ "$old_name" != "$new_name" ]]; then
        mv -v "$old_name" "$new_name"
    fi
}

# Function to rename a file/directory with upper case letters
rename_uppercase() {
    local old_name="$1"
    local new_name=$(echo "$old_name" | tr '[:lower:]' '[:upper:]')
    if [[ "$old_name" != "$new_name" ]]; then
        mv -v "$old_name" "$new_name"
    fi
}

# Main program
echo "Before renaming:"
ls -R

echo "Renaming files/directories to lowercase:"
for item in *; do
    if [[ -d $item || -f $item ]]; then
        rename_lowercase $item
    fi
done

```

```
rename_uppercase "$item"

elif [[ -d "$item" ]]; then

    cd "$item"

    for subitem in *; do

        rename_uppercase "$subitem"

    done

    cd ..

    rename_uppercase "$item"

done

echo "After renaming:"

ls -R
```

```

hubham@DESKTOP-670QJ40:~/shell$ vi q13.sh
hubham@DESKTOP-670QJ40:~/shell$ chmod u+x q13.sh
hubham@DESKTOP-670QJ40:~/shell$ ./q13.sh
before renaming:
:
1.sh a2.sh a3.sh a4.sh fifth first first.sh for.sh fourth q1.sh q10.sh q11.sh q12.sh q13.sh q2.sh q3.sh q4.sh
.sh q6.sh q7.sh q8.sh q9.sh second second.sh third third.sh

/fifth:
/first:
/fourth:
/second:
/third:
renaming files/directories to lowercase:
renaming files/directories to uppercase:
renamed 'a1.sh' -> 'A1.SH'
renamed 'a2.sh' -> 'A2.SH'
renamed 'a3.sh' -> 'A3.SH'
renamed 'a4.sh' -> 'A4.SH'
renamed 'fifth' -> 'FIFTH'
renamed 'first' -> 'FIRST'
renamed 'first.sh' -> 'FIRST.SH'
renamed 'for.sh' -> 'FOR.SH'
renamed 'fourth' -> 'FOURTH'
renamed 'q1.sh' -> 'Q1.SH'
renamed 'q10.sh' -> 'Q10.SH'
renamed 'q11.sh' -> 'Q11.SH'
renamed 'q12.sh' -> 'Q12.SH'
renamed 'q13.sh' -> 'Q13.SH'
renamed 'q2.sh' -> 'Q2.SH'
renamed 'q3.sh' -> 'Q3.SH'
renamed 'q4.sh' -> 'Q4.SH'
renamed 'q5.sh' -> 'Q5.SH'
renamed 'q6.sh' -> 'Q6.SH'
renamed 'q7.sh' -> 'Q7.SH'
renamed 'q8.sh' -> 'Q8.SH'
renamed 'q9.sh' -> 'Q9.SH'
renamed 'second' -> 'SECOND'
renamed 'second.sh' -> 'SECOND.SH'
renamed 'third' -> 'THIRD'
renamed 'third.sh' -> 'THIRD.SH'
fter renaming:
:
1.SH A2.SH A3.SH A4.SH FIFTH FIRST FIRST.SH FOR.SH FOURTH Q1.SH Q10.SH Q11.SH Q12.SH Q13.SH Q2.SH Q3.SH Q4.SH

```

```

#!/bin/bash

usernames=$(awk -F: '{ print $1, length($1) }' /etc/passwd)

longest=$(echo "$usernames" | sort -k2 -rn | head -n1)
shortest=$(echo "$usernames" | sort -k2 -rn | tail -n1)

longest_username=$(echo "$longest" | awk '{ print $1 }')
longest_length=$(echo "$longest" | awk '{ print $2 }')
shortest_username=$(echo "$shortest" | awk '{ print $1 }')
shortest_length=$(echo "$shortest" | awk '{ print $2 }')

echo "The user with longest name is $longest_username with length $longest_length"
echo "The user with shortest name is $shortest_username with length $shortest_length"

```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q14.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q14.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q14.sh
The user with longest name is systemd-timesync with length 16
The user with shortest name is lp with length 2
```

```
#!/bin/bash

echo "How many random passwords would you like to generate?"
read num_passwords

for ((i=1;i<=$num_passwords;i++)); do
    password=$(openssl rand -base64 6 | tr -d "=/+")
    echo "Password $i: $password"
done
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q15.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q15.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q15.sh
How many random passwords would you like to generate?
2
Password 1: WBTPDlga
Password 2: IH8pEyTL
```

```

#!/bin/bash

hour=$(date +%H)
day=$(date +%A)
date=$(date +%B" "%d", "%Y)

if [ $hour -ge 5 -a $hour -lt 12 ]; then
    greeting="Good morning!"
elif [ $hour -ge 12 -a $hour -lt 18 ]; then
    greeting="Good afternoon!"
else
    greeting="Good evening!"
fi

echo "$greeting Today is $day, $date."

```

```

shubham@DESKTOP-670QJ40:~/shell$ vi q16.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q16.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q16.sh
Good afternoon! Today is Thursday, April 20, 2023.
shubham@DESKTOP-670QJ40:~/shell$
```

```

#!/bin/bash

echo -n "Enter the file name: "
read file_name

if [ ! -f $file_name ]; then
    echo "File does not exist"
    exit 1
fi
echo 1 = To Upper Case
echo 2 = To Lower Case

echo -n "Convert to uppercase or lowercase? Please enter your choice"
read choice

case $choice in
    1)
        tr '[[:lower:]]' '[[:upper:]]' < $file_name
        ;;
    2)
        tr '[[:upper:]]' '[[:lower:]]' < $file_name
        ;;
    *)
        echo "Invalid choice"
        exit 1
esac
```

```
name
```

```
        echo "Invalid choice"  
        exit 1  
    ;;  
esac
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q17.sh  
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q17.sh  
shubham@DESKTOP-670QJ40:~/shell$ ./q17.sh  
Enter the file name: A2.SH  
1 = To Upper Case  
2 = To Lower Case  
Convert to uppercase or lowercase? Please enter your choice1  
#!/BIN/BASH  
  
IFS=':' READ -RA DIRS <<< "$PATH"  
  
FOR DIR IN "${DIRS[@]}";  
DO  
  
    NUM_EXEC=$(FIND "$DIR" -TYPE F -PERM +X | WC -L)  
  
    ECHO "DIRECTORY: $DIR"  
    ECHO "NUMBER OF EXECUTABLE FILES: $NUM_EXEC"  
    ECHO ""  
  
DONE
```

```
shubham@DESKTOP-670QJ40:~/shell$ ./q17.sh
Enter the file name: A2.SH
1 = To Upper Case
2 = To Lower Case
Convert to uppercase or lowercase? Please enter your choice2
#!/bin/bash

IFS=':' read -ra dirs <<< "$path"

for dir in "${dirs[@]}";
do

    num_exec=$(find "$dir" -type f -perm +x | wc -l)

    echo "directory: $dir"
    echo "number of executable files: $num_exec"
    echo ""

done
```

```
#!/bin/bash

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

filename="$1"

if [[ ! -r "$filename" ]]; then
    echo "Error: $filename is not a readable file"
    exit 1
fi

content=$(cat "$filename")

if [[ "$content" == "${content,,}" ]]; then
    echo "${content^^}" > "$filename"
    echo "Converted $filename to uppercase"
elif [[ "$content" == "${content^^,,}" ]]; then
    echo "${content,,}" > "$filename"
    echo "Converted $filename to lowercase"
else
    echo "Error: $filename contains mixed case content"
    exit 1
fi
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q18.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q18.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q18.sh
Usage: ./q18.sh [filename]
shubham@DESKTOP-670QJ40:~/shell$ ./q18.sh A2.SH
Error: A2.SH contains mixed case content
shubham@DESKTOP-670QJ40:~/shell$ cat demo.txt
cat: demo.txt: No such file or directory
shubham@DESKTOP-670QJ40:~/shell$ cat > demo.txt
dsajdshd ndsbshds dhsshd
ndhfdssfs hdhsdgk
jfjdkshf dfdfshfkj fjdskh
jdfjkf mdfbdsk sdfjkds
dbndbds dsnbndssh
shubham@DESKTOP-670QJ40:~/shell$ ./q18.sh demo.txt
Converted demo.txt to uppercase
shubham@DESKTOP-670QJ40:~/shell$ cat demo.txt
DSAJDSDHD NDSBSHDS DHSDSSH
DHDFDSFS HDHSDGK
JFJDKSHF DFDFSHFKJ FJDSKH
JDFJKF MDFBDSK SDFJKDS
DBNDBDS DSBNBND
```

```
#!/bin/bash

while true;
do
    echo "hello" | cat
done
```

```
#!/bin/bash

print_arguments()

    if [[ $# -gt 0 ]];
        then
            echo "$1"
            shift
            print_arguments "$@"
        fi
    }

    print_arguments "$@" -
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q20.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q20.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q20.sh
shubham@DESKTOP-670QJ40:~/shell$ vi q20.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q20.sh abc def ghi
abc
def
ghi
shubham@DESKTOP-670QJ40:~/shell$
```

```
#!/bin/bash

if [ $# -eq 0 ];
then
        echo "Please provide a file system or mount point as an argument"
        exit 1
fi

if mountpoint -q -- "$1";
then
        echo "$1 is a currently mounted file system"
else
        echo "$1 is not a currently mounted file system"
fi
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q21.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q21.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q20.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q21.sh
Please provide a file system or mount point as an argument
shubham@DESKTOP-670QJ40:~/shell$ ./q21.sh /dev/sda1
/dev/sda1 is not a currently mounted file system
shubham@DESKTOP-670QJ40:~/shell$ ./q21.sh /dev/sda8
/dev/sda8 is not a currently mounted file system
shubham@DESKTOP-670QJ40:~/shell$
```

```
#!/bin/bash

For dir in "$@"
do

    if [ -d "$dir" ]
    then

        echo "Contents of $dir:"
        ls -l "$dir"
    else

        echo "$dir is not a directory"

    fi

done
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q22.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q22.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q22.sh
shubham@DESKTOP-670QJ40:~/shell$ ls
A1.SH  A4.SH  FIRST.SH  Q1.SH  Q12.SH  Q3.SH  Q6.SH  Q9.SH      THIRD      q13.sh  q16.sh  q19.sh  q22.sh
A2.SH  FIFTH   FOR.SH   Q10.SH  Q13.SH  Q4.SH  Q7.SH  SECOND    THIRD.SH  q14.sh  q17.sh  q20.sh
A3.SH  FIRST   FOURTH   Q11.SH  Q2.SH   Q5.SH  Q8.SH  SECOND.SH demo.txt  q15.sh  q18.sh  q21.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q22.sh FIFTH
Contents of FIFTH:
total 0
```

```
#!/bin/bash

if [ $# -eq 0 ]
then
    echo "Usage: bash $0 <directory>"
    exit 1
fi

echo "Original file permissions:"
ls -l "$1"

chmod go-rwx "$1"/*
chmod go-rwx "$1"

echo "New file permissions:"
ls -l "$1"
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q23.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q23.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q23.sh
Usage: bash ./q23.sh <directory>
shubham@DESKTOP-670QJ40:~/shell$ ./q23.sh FIFTH
Original file permissions:
total 0
chmod: cannot access 'FIFTH/*': No such file or directory
New file permissions:
./q23.sh: line 20: ls -l FIFTH: command not found
shubham@DESKTOP-670QJ40:~/shell$ ./q23.sh SECOND
Original file permissions:
total 0
chmod: cannot access 'SECOND/*': No such file or directory
New file permissions:
./q23.sh: line 20: ls -l SECOND: command not found
shubham@DESKTOP-670QJ40:~/shell$ ./q23.sh FIRST
Original file permissions:
total 0
chmod: cannot access 'FIRST/*': No such file or directory
New file permissions:
./q23.sh: line 20: ls -l FIRST: command not found
shubham@DESKTOP-670QJ40:~/shell$ cd FIFTH
shubham@DESKTOP-670QJ40:~/shell/FIFTH$ cat > demo1.txt
yeghs hjfgfih hfghsd
sfjhdsf fhd sdfhd
slfhdsjkfh fbdhb
shubham@DESKTOP-670QJ40:~/shell/FIFTH$ chmod u+x demo1.txt
shubham@DESKTOP-670QJ40:~/shell/FIFTH$ cd ..
```

```
shubham@DESKTOP-670QJ40:~/shell/FIFTH$ cd ..
shubham@DESKTOP-670QJ40:~/shell$ ./q23.sh FIFTH
Original file permissions:
total 4
-rwxr--r-- 1 shubham shubham 56 Apr 20 17:05 demo1.txt
```

```
#!/bin/bash

if_output=$(df -B1)

while read -r line;
do

    if [[ $line == Filesystem* ]];
        then
            continue
        fi
                fs=$(echo "$line" | awk '{print $1}')
        used=$(echo "$line" | awk '{print $5}')
        available=$(echo "$line" | awk '{print $4}')
                used_percentage=$(echo "scale=2; $used * 100 / ($used + $available)" | bc)
                available_percentage=$(echo "scale=2; $available * 100 / ($used + $available)" | bc)

        if (( available_percentage < 10 ));

    then
        echo "File-system $fs has less than 10% free space available."
        fi

```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q24.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q24.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q24.sh
./q24.sh: line 18: bc: command not found
./q24.sh: line 19: bc: command not found
file-system Filesystem has less than 10% free space available.
./q24.sh: line 18: bc: command not found
./q24.sh: line 19: bc: command not found
file-system none has less than 10% free space available.
./q24.sh: line 18: bc: command not found
./q24.sh: line 19: bc: command not found
file-system none has less than 10% free space available.
./q24.sh: line 18: bc: command not found
./q24.sh: line 19: bc: command not found
file-system none has less than 10% free space available.
./q24.sh: line 18: bc: command not found
./q24.sh: line 19: bc: command not found
file-system /dev/sdc has less than 10% free space available.
./q24.sh: line 18: bc: command not found
./q24.sh: line 19: bc: command not found
file-system none has less than 10% free space available.
./q24.sh: line 18: bc: command not found
./q24.sh: line 19: bc: command not found
file-system rootfs has less than 10% free space available.
./q24.sh: line 18: bc: command not found
./q24.sh: line 19: bc: command not found
file-system none has less than 10% free space available.
./q24.sh: line 18: bc: command not found
./q24.sh: line 19: bc: command not found
file-system none has less than 10% free space available.
./q24.sh: line 18: bc: command not found
```

```
#!/bin/bash
count=0
while IFS=: read -r user_id rest
do
    if [[ $user_id -ge 500 && $user_id -le 10000 ]]
        then
            count=$((count+1))
            fi
done < /etc/passwd
echo "Number of users with user IDs between 500 and 1000
$: $count"
```

```
hubham@DESKTOP-670QJ40:~/shell$ vi q25.sh
hubham@DESKTOP-670QJ40:~/shell$ chmod u+x q25.sh
hubham@DESKTOP-670QJ40:~/shell$ ./q25.sh
umber of users with user IDs between 500 and 10000: 0
```

```
#!/bin/bash

IFS=':' read -ra dirs <<< "$PATH"

for dir in "${dirs[@]}";
do

num_exec=$(find "$dir" -type f -perm +x | wc -l)

echo "Directory: $dir"
echo "Number of executable files: $num_exec"
echo ""

done
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q26.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q26.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q26.sh
Directory: /usr/local/sbin
Number of executable files: 0

Directory: /usr/local/bin
Number of executable files: 0

Directory: /usr/sbin
Number of executable files: 0

Directory: /usr/bin
Number of executable files: 0

Directory: /sbin
Number of executable files: 0

Directory: /bin
Number of executable files: 0

Directory: /usr/games
Number of executable files: 0

Directory: /usr/local/games
Number of executable files: 0

Directory: /usr/lib/wsl/lib
Number of executable files: 0
```

```
#!/bin/bash

echo "Enter the username to search for: "
read username

if grep -q "^$username:" /etc/passwd ; then
    echo "User $username found in the system."
else
    echo "User $username not found in the system."
fi
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q27.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q27.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q27.sh
Enter the username to search for:
shubham
User shubham found in the system.
```

```
shubham@DESKTOP-670QJ40:~/shell$ ./q27.sh
Enter the username to search for:
raj
User raj not found in the system.
```

```
#!/bin/bash

echo "Enter your weight in kilograms:"
read weight

echo "Enter your height in meters:"
read height

bmi=$(echo "scale=2; $weight / ($height * $height)" | bc)

echo "Your BMI is $bmi"

if (( $(echo "$bmi < 18.5" | bc -l) )); then
    echo "You are underweight."
    elif (( $(echo "$bmi < 25" | bc -l) )); then
        echo "You have a healthy weight."
        elif (( $(echo "$bmi < 30" | bc -l) )); then
            echo "You are overweight."
            else
                echo "Y
ou are obese."
fi
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q28.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q28.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q28.sh
```

```
sudo] password for shubham:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  bc  
    upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
  Need to get 87.6 kB of archives.  
After this operation, 220 kB of additional disk space will be used.  
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 bc amd64 1.07.1-3build1  
  Etched: 87.6 kB in 2s (53.4 kB/s)  
Selecting previously unselected package bc.  
Reading database ... 24301 files and directories currently installed.)  
Preparing to unpack .../bc_1.07.1-3build1_amd64.deb ...  
Unpacking bc (1.07.1-3build1) ...  
Setting up bc (1.07.1-3build1) ...  
Processing triggers for install-info (6.8-4build1) ...  
Processing triggers for man-db (2.10.2-1) ...  
shubham@DESKTOP-670QJ40:~/shell$ ./q28.sh  
Enter your weight in kilograms:  
8  
Enter your height in meters:  
.63  
Your BMI is 25.66  
You are overweight.  
shubham@DESKTOP-670QJ40:~/shell$ ./q28.sh  
Enter your weight in kilograms:  
8  
Enter your height in meters:  
.7  
Your BMI is 23.52  
You have a healthy weight.
```

```
#!/bin/bash

echo "1. Currently logged users"
echo "2. Your shell directory"
echo "3. Home directory"
echo "4. OS name & version"
echo "5. Current working directory"
echo "6. Number of users logged in"
echo "7. Show all available shells in your system"
echo "8. Hard disk information"
echo "9. CPU information"
echo "10. Memory information"
echo "11. File system information"
echo "12. Currently running process"

read -p "Enter the option: " choice

case $choice in
    1) who;;
    2) echo "Your shell directory is $SHELL";;
    3) echo "Your home directory is $HOME";;
    4) uname -a;;
    5) echo "Your current working directory is $PWD";;
    6) who | wc -l;;
    7) cat /etc/shells;;
    8) df -h;;
    9) lscpu;;
    10) free -m;;
    11) df -T;;
    12) ps -ef;;
    *) echo "Invalid option";;
esac
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q29.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q29.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q29.sh
1. Currently logged users
2. Your shell directory
3. Home directory
4. OS name & version
5. Current working directory
6. Number of users logged in
7. Show all available shells in your system
8. Hard disk information
9. CPU information
10. Memory information
11. File system information
12. Currently running process
Enter the option: 3
Your home directory is /home/shubham
```

```
#!/bin/bash

if [ $# -eq 0 ]
then
    echo "Error: new directory name not provided"
    exit 1
fi

current_dir=$(basename "$PWD")

if [ "$current_dir" == "$1" ]
then
    echo "Error: current directory name and new name are the same"
    exit 1
fi

mv "$PWD" "$(dirname "$PWD")/$1"
echo "Directory renamed from $current_dir to $1"
```

```
shubham@DESKTOP-670QJ40:~/shell$ vi q29.sh
shubham@DESKTOP-670QJ40:~/shell$ vi q30.sh
shubham@DESKTOP-670QJ40:~/shell$ chmod u+x q30.sh
shubham@DESKTOP-670QJ40:~/shell$ ./q30.sh
Error: new directory name not provided
shubham@DESKTOP-670QJ40:~/shell$ ./q30.sh FIFTH
Directory renamed from shell to FIFTH
```

```
#!/bin/bash

if [ $# -eq 0 ]
    then
        echo "Error: new prefix not provided"
        exit 1
fi

for file in *.png
do

    name=$(basename "$file" .png)

    newname="${name#*_}"

    newname="$1_"$newname".png"

    mv "$file" "$newname"
done

echo "All .jpg files renamed with prefix $1"
```

```
shubham@DESKTOP-670QJ40:~/FIFTH$ vi q31.sh
shubham@DESKTOP-670QJ40:~/FIFTH$ chmod u+x q31.sh
shubham@DESKTOP-670QJ40:~/FIFTH$ ./q31.sh FIRST.SH
mv: cannot stat '*.png': No such file or directory
All .jpg files renamed with prefix FIRST.SH
```

```
#!/bin/bash

if [ $# -lt 3 ]
then
    echo "Error: missing arguments"
    echo "Usage: $0 <file> <start line> <number of lines>"
    exit 1
fi

file="$1"
start="$2"
num="$3"

tail -n +$start "$file" | head -n $num
```

```
shubham@DESKTOP-670QJ40:~/FIFTH$ vi q32.sh
shubham@DESKTOP-670QJ40:~/FIFTH$ chmod u+x q32.sh
shubham@DESKTOP-670QJ40:~/FIFTH$ ./q32.sh demo.txt 2
Error: missing arguments
Usage: ./q32.sh <file> <start line> <number of lines>
shubham@DESKTOP-670QJ40:~/FIFTH$ ./q32.sh demo.txt 2 2
HDHFDSFS HDHSDGK
JFJDKSHF DFDSHFKJ FJDSKH
```

```
#!/bin/bash

echo "swp files found:"
find . -name "*.swp" -print

read -p "Do you want to delete all .swp files? (y/n)" confirm

if [[ $confirm == "y" ]]; then
    find . -name "*.swp" -type f -delete
    echo "All .swp files deleted!"
else
    echo "No files were deleted."
fi
```

```
shubham@DESKTOP-670QJ40:~/FIFTH$ vi q33.sh
shubham@DESKTOP-670QJ40:~/FIFTH$ chmod u+x q33.sh
shubham@DESKTOP-670QJ40:~/FIFTH$ ./q33.sh demo.txt
swp files found:
Do you want to delete all .swp files? (y/n)y
All .swp files deleted!
```