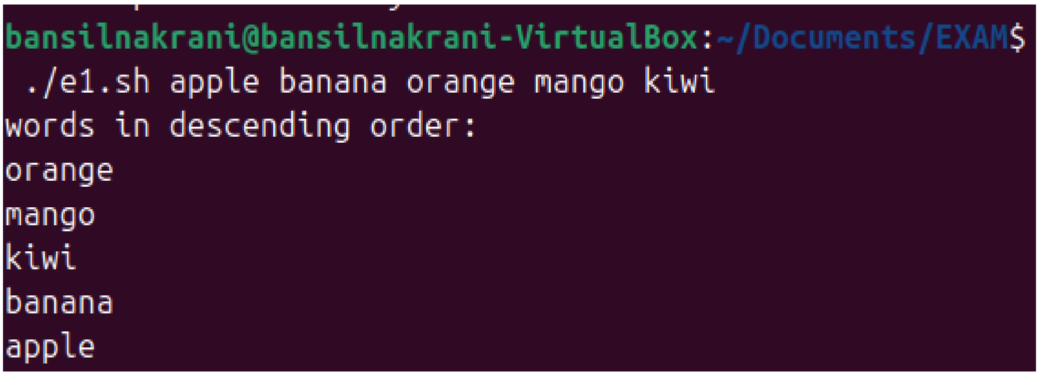
SSP PRACTICAL IMP QUESTION

1. **Write a shell script program which accepts 5 different words from the command line and store them in a text file and display words in**

```
if [ "$#" -ne 5 ]; then
   echo "Please provide exactly 5 words."
   exit 1
fi


echo "$@" > words.txt


echo "words in descending order:"
echo "$@" | tr ' ' '\n' | sort -r
```

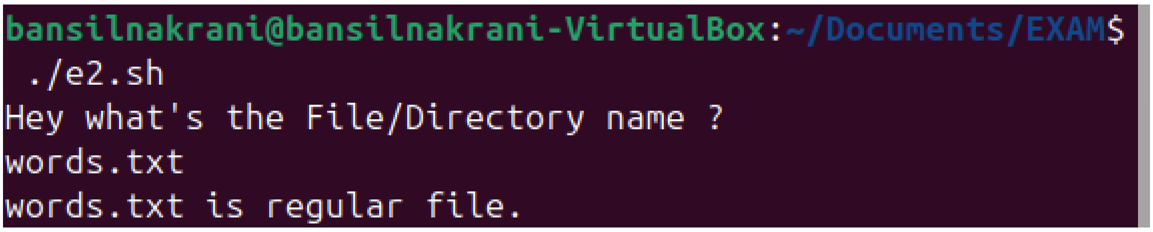


```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM$
 ./e1.sh apple banana orange mango kiwi
words in descending order:
orange
mango
kiwi
banana
apple
```

SSP PRACTICAL IMP QUESTION

2. **Write a Shell Bash Script for evaluate the status of a file/directory.**

```
echo "Hey what's the File/Directory name ?"
read FILE


if [ -e "$FILE" ]; then
        if [ -f "$FILE" ]; then
                echo "$FILE is regular file."
        elif [ -d "$FILE" ]; then
                echo "$FILE is a directory file."
        elif [ -r "$FILE" ]; then
                echo "$FIEL is a readable."
        elif [ -w "$FILE" ]; then
                echo "$FILE is a writable."
        elif [ -x "$FILE" ]; then
                echo "$FILE is executable/searchable."
        fi
else
        echo "$FILE does not exist."
fi
```

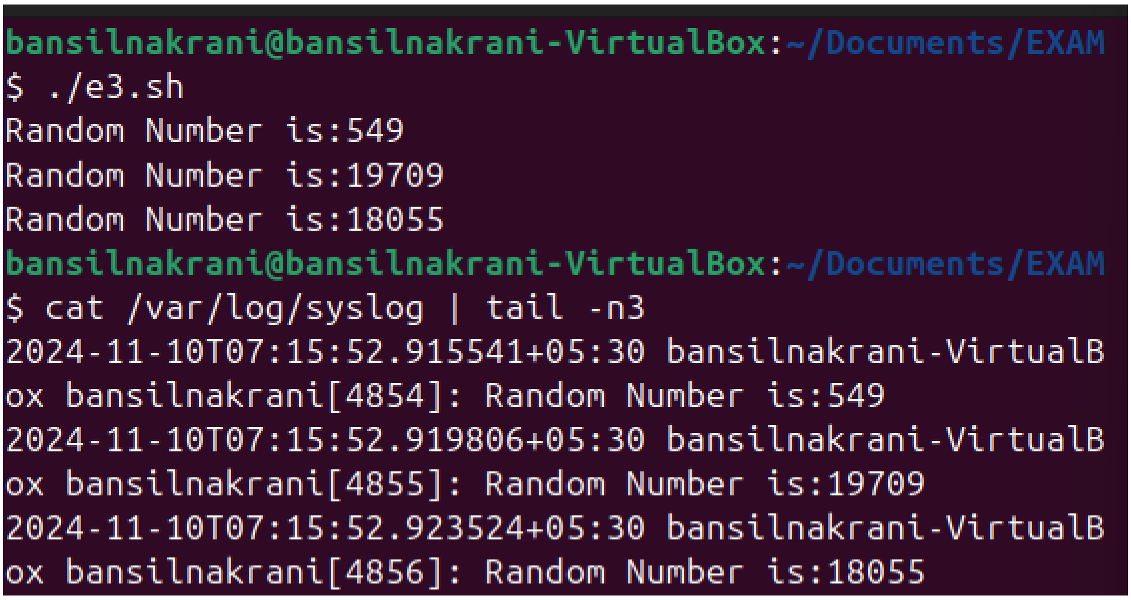


```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM$
 ./e2.sh
Hey what's the File/Directory name ?
words.txt
words.txt is regular file.
```

SSP PRACTICAL IMP QUESTION

3. **Modify the previous script to that it uses a logging function. Additionally, tag each syslog message with "randomly" and include process ID. Generate a 3 random numbers.**

```
function logging() {
  MESSAGE=$@
  SET_MESSAGE="Random  Number is:$MESSAGE"
  echo "$SET_MESSAGE"
  logger -i -p user.info "$SET_MESSAGE"
}
logging $RANDOM
logging $RANDOM
logging $RANDOM
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM
$ ./e3.sh
Random Number is:549
Random Number is:19709
Random Number is:18055
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM
$ cat /var/log/syslog | tail -n3
2024-11-10T07:15:52.915541+05:30 bansilnakrani-VirtualB
ox bansilnakrani[4854]: Random Number is:549
2024-11-10T07:15:52.919806+05:30 bansilnakrani-VirtualB
ox bansilnakrani[4855]: Random Number is:19709
2024-11-10T07:15:52.923524+05:30 bansilnakrani-VirtualB
ox bansilnakrani[4856]: Random Number is:18055
```

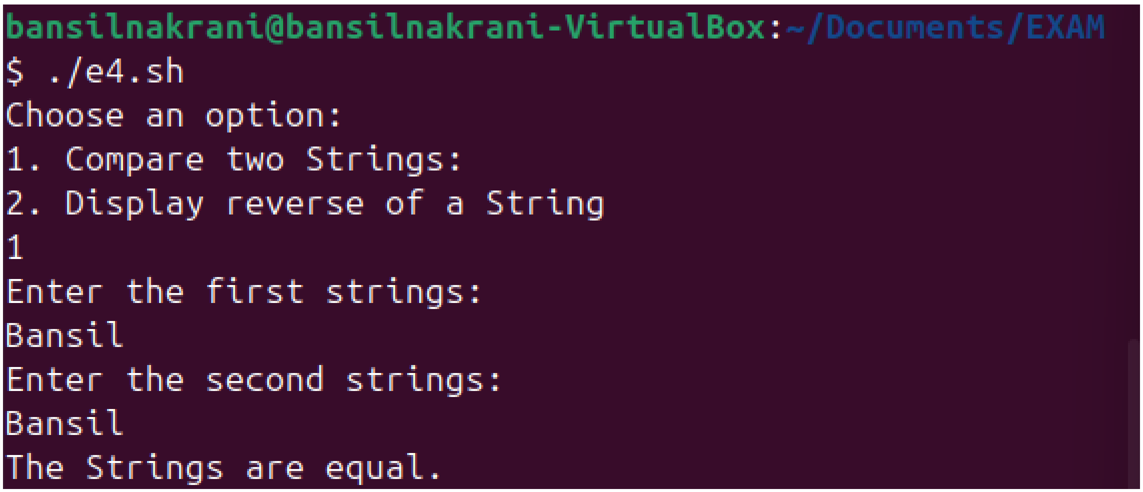
SSP PRACTICAL IMP QUESTION

4. **Write a shell script which provide following facilities**

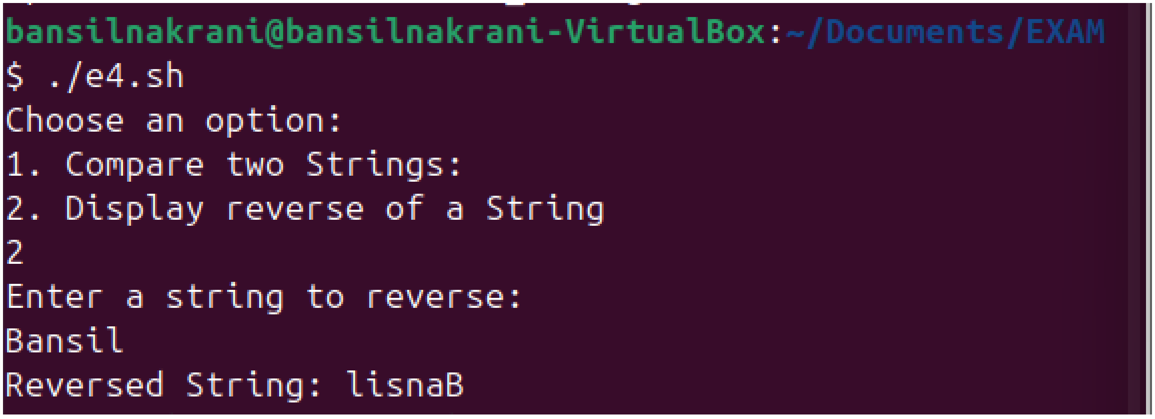
   a. Comparison of two strings.

   b. Display reverse string.

```
ompare_strings() {
        echo "Enter the first strings:"
        read string1
        echo "Enter the second strings:"
        read string2

        if [ "$string1" = "$string2" ]; then
                echo "The Strings are equal."
        else
                echo "The Strings are not equal."
        fi
}

reverse_string() {
        echo "Enter a string to reverse:"
        read string

        reversed_string=$(echo "$string" | rev)
        echo "Reversed String: $reversed_string"
}

echo "Choose an option:"
echo "1. Compare two Strings:"
echo "2. Display reverse of a String"
read choice

case $choice in
        1)
           compare_strings
           ;;
        2)
           reverse_string
        ;;
      *)
        echo "Invalid option. Please choose 1 or 2."
        ;;
esac
```



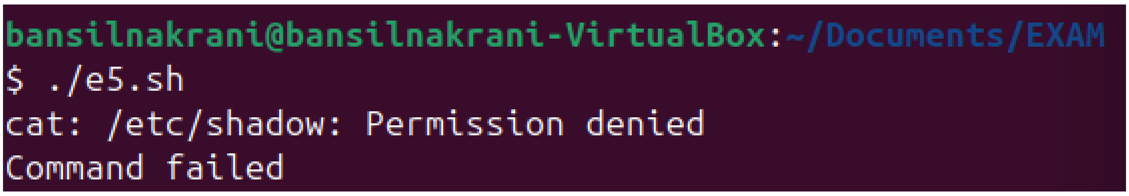
```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM
$ ./e4.sh
Choose an option:
1. Compare two Strings:
2. Display reverse of a String
1
Enter the first strings:
Bansil
Enter the second strings:
Bansil
The Strings are equal.
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM
$ ./e4.sh
Choose an option:
1. Compare two Strings:
2. Display reverse of a String
2
Enter a string to reverse:
Bansil
Reversed String: lisnaB
```

5. **Write a script that executes the command "cat/etc/shadow". If the command return a 0 exit status, report "command succeeded" and exit with a 0 exit status. If the command returns a non-zero exit status, report "Command failed" and exit with a 1 exit status.**

```
cat /etc/shadow

if [ "$?" -eq "0" ]; then
   echo "Command succeeded"
   exit 0
else
   echo "Command failed"
   exit 1
fi
```

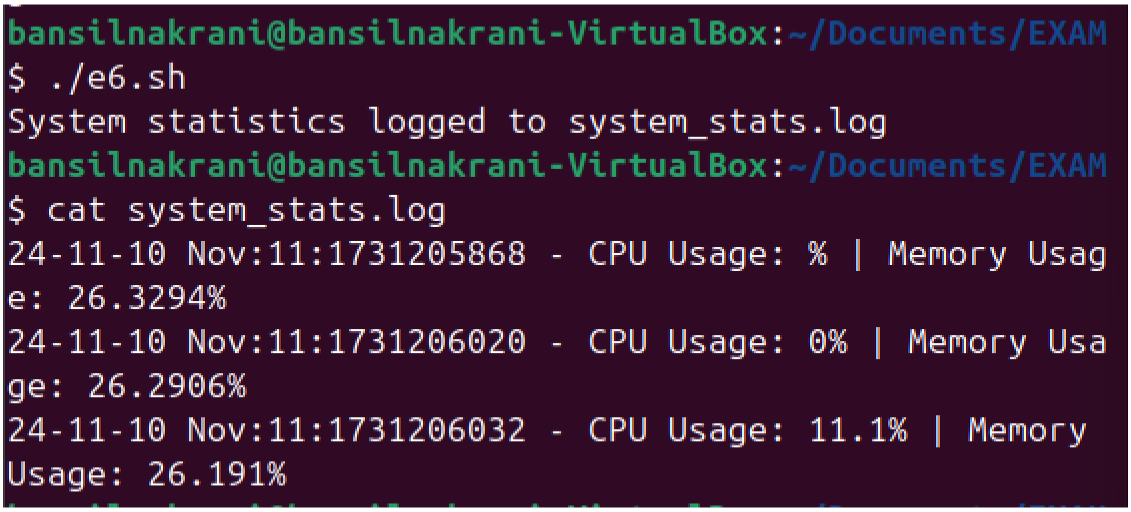


```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM
$ ./e5.sh
cat: /etc/shadow: Permission denied
Command failed
```

SSP PRACTICAL IMP QUESTION

6. **Develop a script that checks CPU and memory usage and logs the statistics to a file, including a timestamp.**

```
LOG_FILE="system_stats.log"

timestamp=$(date '+%y-%m-%d %h:%m:%s')

cpu_usage=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}')

mem_usage=$(free | grep Mem | awk '{print $3/$2 * 100.0}')

echo "$timestamp - CPU Usage: $cpu_usage% | Memory Usage: $mem_usage%" >> "$LOG_FILE"

echo "System statistics logged to $LOG_FILE"
```

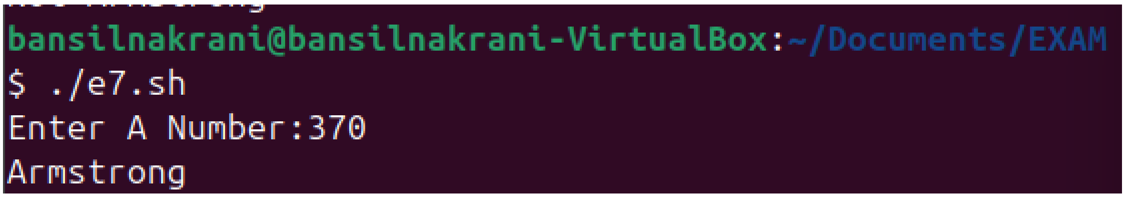


```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM
$ ./e6.sh
System statistics logged to system_stats.log
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM
$ cat system_stats.log
24-11-10 Nov:11:1731205868 - CPU Usage: % | Memory Usag
e: 26.3294%
24-11-10 Nov:11:1731206020 - CPU Usage: 0% | Memory Usa
ge: 26.2906%
24-11-10 Nov:11:1731206032 - CPU Usage: 11.1% | Memory
Usage: 26.191%
```

SSP PRACTICAL IMP QUESTION

7. **Write a Shell Bash Script for check if a provided number is Armstrong or not.**

```
echo -n "Enter A Number:"
read -r n

arm=0
temp=$n
num_digits=${#n}

while [ "$n" -ne 0 ];
   do
        r=$((n % 10))
        arm=$((arm + r**num_digits))
        n=$((n / 10))
   done


if [ "$arm" -eq "$temp"  ]; then
        echo "Armstrong"
else
        echo "Not Armstrong"
fi
```

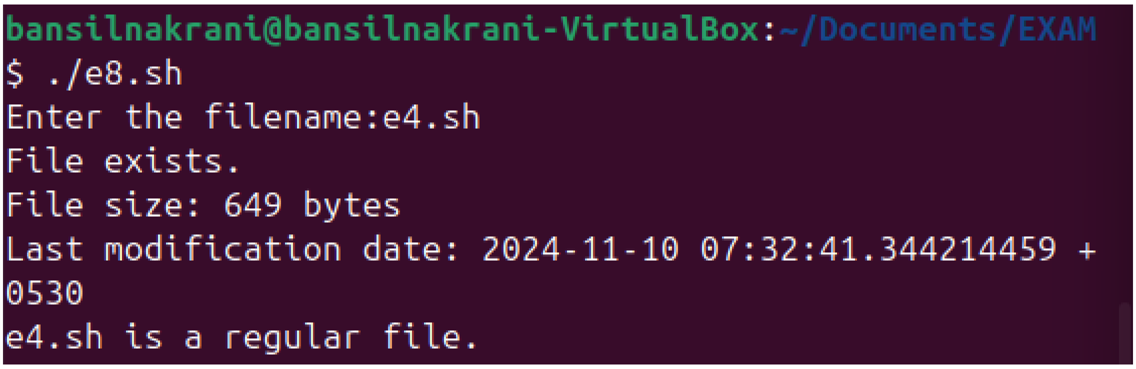


```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM
$ ./e7.sh
Enter A Number:370
Armstrong
```

SSP PRACTICAL IMP QUESTION

8. **Write a script that accepts a filename as an argument and checks if the file exists. If it exists, display its size, modification date, and whether it’s a regular file or a directory.**

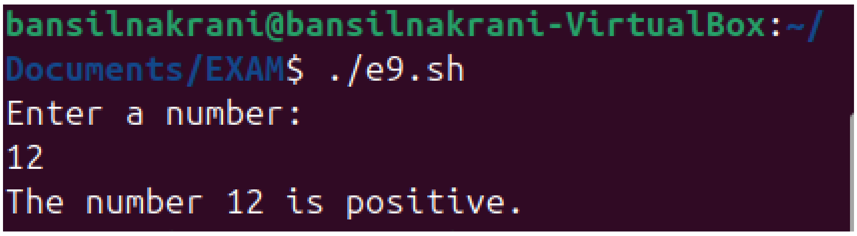
```
echo -n "Enter the filename:"
read -r filename

if [ -e "$filename" ]; then
  echo "File exists."

  filesize=$(stat -c %s "$filename")
  echo "File size: $filesize bytes"

  mod_date=$(stat -c %y "$filename")
  echo "Last modification date: $mod_date"

  if [ -f "$filename" ]; then
        echo "$filename is a regular file."
  elif [ -d "$filename" ]; then
        echo "$filename is a directory."
  else
        echo "$filename is neither a regular file nor a directory."
  fi
else
  echo "file does not exist."
  exit 2
fi
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM
$ ./e8.sh
Enter the filename:e4.sh
File exists.
File size: 649 bytes
Last modification date: 2024-11-10 07:32:41.344214459 +
0530
e4.sh is a regular file.
```

9. Write a shell script to check whether the number is positive or negative. Also, display an error message if the number entered is Zero.

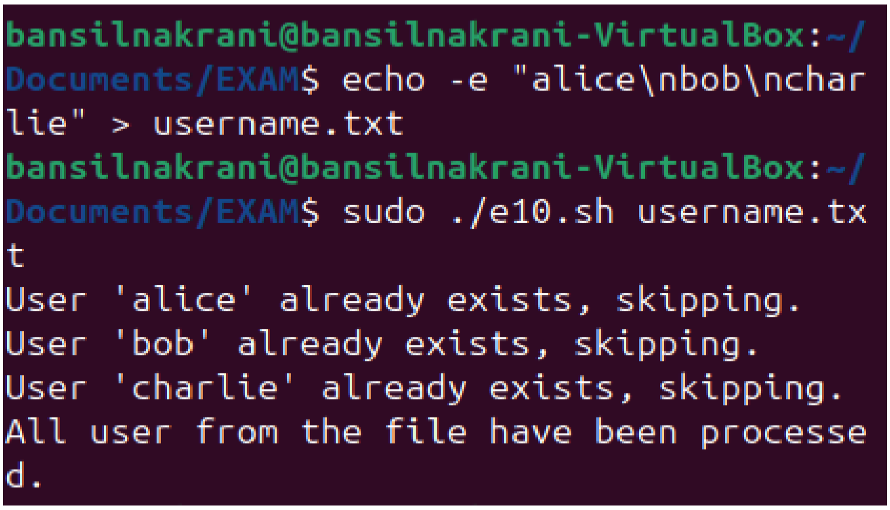
```
echo "Enter a number:"
read -r num

if [ "$num" -eq 0 ]; then
        echo "Error: You entered zero. Please enter a non zero number."
elif [ "$num" -gt 0 ]; then
        echo "The number $num is positive."
else
        echo "The number $num is negative."
fi
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/
Documents/EXAM$ ./e9.sh
Enter a number:
12
The number 12 is positive.
```

10. Write a script that reads a list of usernames from a file and creates each user on the system, setting a default password for each.

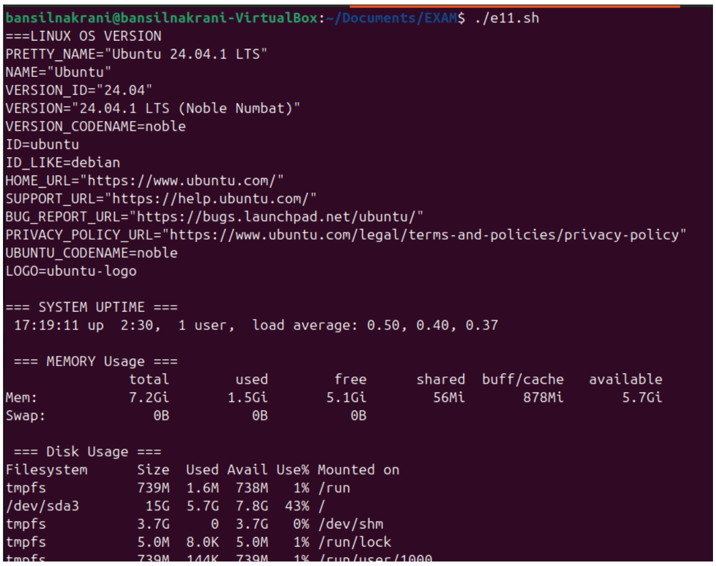
```
if [ -z "$1" ]; then
        echo "Usage: $0 <username_file>"
        exit 1
fi

DEFAULT_PASSWORD="Password123!"

while read -r username; do
        if [ -n "$username" ]; then
                if id "$username" &>/dev/null; then
                        echo "User '$username' already exists, skipping."
                else
                        useradd "$username"
                        echo "$username:$DEFAULT_PASSWORD" | chpasswd
                        echo "User '$username' created with default
password."
                fi
        fi
done < "$1"

echo "All user from the file have been processed."
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/
Documents/EXAM$ echo -e "alice\nbob\nchar
lie" > username.txt
bansilnakrani@bansilnakrani-VirtualBox:~/
Documents/EXAM$ sudo ./e10.sh username.tx
t
User 'alice' already exists, skipping.
User 'bob' already exists, skipping.
User 'charlie' already exists, skipping.
All user from the file have been processe
d.
```

11.Write a Shell Bash Script for report server related information with linux os version.

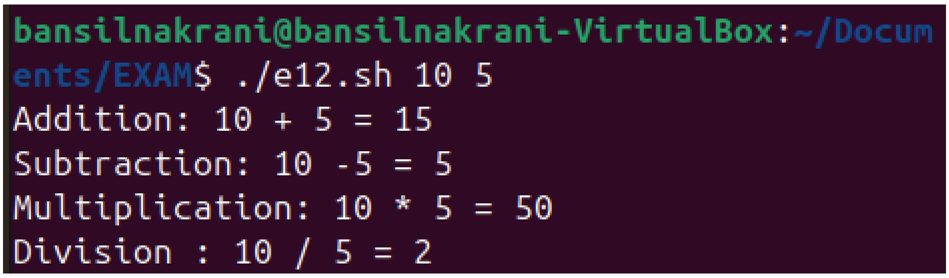
```
echo "===LINUX OS VERSION"
cat /etc/os-release

echo -e "\n=== SYSTEM UPTIME ==="
uptime

echo -e "\n === MEMORY Usage ==="
free -h

echo -e "\n === Disk Usage ==="
df -h

echo -e "\n === CPU Information ==="
lscpu | grep -E '^Model name|^CPU\(s\):|^Architecture|^CPU  MHz'

echo -e "\n === Network Information (IP Address) ==="
ip -brief address

echo -e "\n === top 5 running process by memory usage ==="
ps -eo pid,comm,%mem  --sort=-%mem  | head -n 6
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM$ ./e11.sh
===LINUX OS VERSION
PRETTY_NAME="Ubuntu 24.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.1 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo

=== SYSTEM UPTIME ===
 17:19:11 up  2:30,  1 user,  load average: 0.50, 0.40, 0.37

 === MEMORY Usage ===
               total        used        free      shared  buff/cache   available
Mem:           7.2Gi       1.5Gi       5.1Gi        56Mi       878Mi       5.7Gi
Swap:             0B          0B          0B

 === Disk Usage ===
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           739M  1.6M  738M   1% /run
/dev/sda3        15G  5.7G  7.8G  43% /
tmpfs           3.7G     0  3.7G   0% /dev/shm
tmpfs           5.0M  8.0K  5.0M   1% /run/lock
tmpfs           739M  144K  739M   1% /run/user/1000
```

12. Write a script that takes two numbers and an operator (+, -, *, /) as arguments and performs the calculation, displaying the result.

```
if [ "$#" -ne 2 ]; then
  echo "Error: please provide 2 number as command line arguments."
  exit 1
fi



num1=$1
num2=$2

sum=$(($num1 + $num2))
difference=$(($num1 - $num2))
product=$(($num1 * $num2))
if [ $num2 -eq 0 ]; then
   echo "Error : division by zero is not allowed."
else
   division=$(($num1/$num2))
fi

echo "Addition: $num1 + $num2 = $sum"
echo "Subtraction: $num1 -$num2 = $difference"
echo "Multiplication: $num1 * $num2 = $product"
echo "Division : $num1 / $num2 = $division"
```

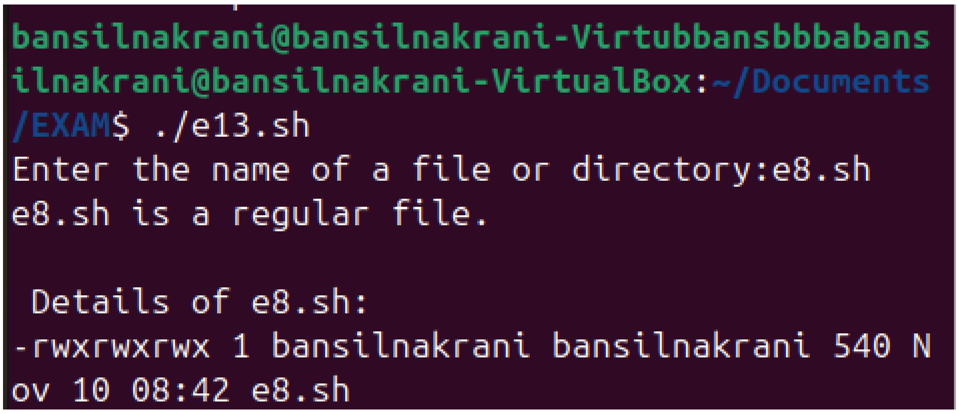


```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM$ ./e12.sh 10 5
Addition: 10 + 5 = 15
Subtraction: 10 -5 = 5
Multiplication: 10 * 5 = 50
Division : 10 / 5 = 2
```

SSP PRACTICAL IMP QUESTION

13. write a shell script that prompts the user for a name of a file or directory and reports if it is a regular file, a directory, or another type of file. Also perform an ls command against the file or directory with the long listing option.

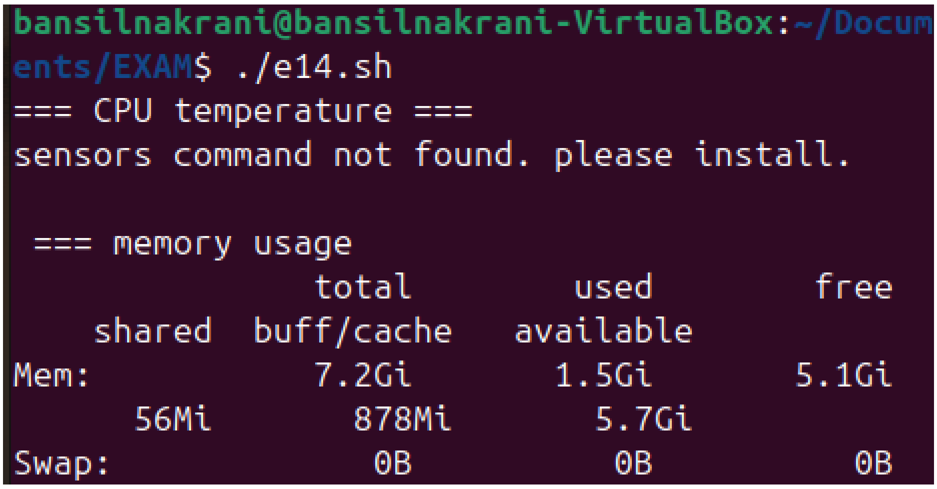
```
echo -n "Enter the name of a file or directory:"
read file

if [ -e "$file" ]; then
        if [ -f "$file" ]; then
                echo "$file is a regular file."
        elif [ -d "$file" ]; then
                echo "$file is a directory."
        else
                echo "$file is of another type."
        fi

        echo -e "\n Details of $file:"
        ls -l "$file"
else
        echo "The file or directory '$file' does not exist."
fi
```



```
bansilnakrani@bansilnakrani-Virtubbansbbbabans
ilnakrani@bansilnakrani-VirtualBox:~/Documents
/EXAM$ ./e13.sh
Enter the name of a file or directory:e8.sh
e8.sh is a regular file.

 Details of e8.sh:
-rwxrwxrwx 1 bansilnakrani bansilnakrani 540 N
ov 10 08:42 e8.sh
```

14. Write a Shell Bash Script to show CPU temperature with memory usage.

```
echo "=== CPU temperature ==="


if command -v sensors > /dev/null; then

        sensors | grep -i 'temp' | head -n 1

else

        echo "sensors command not found. please install."

fi


echo -e "\n === memory usage"

free -h
```



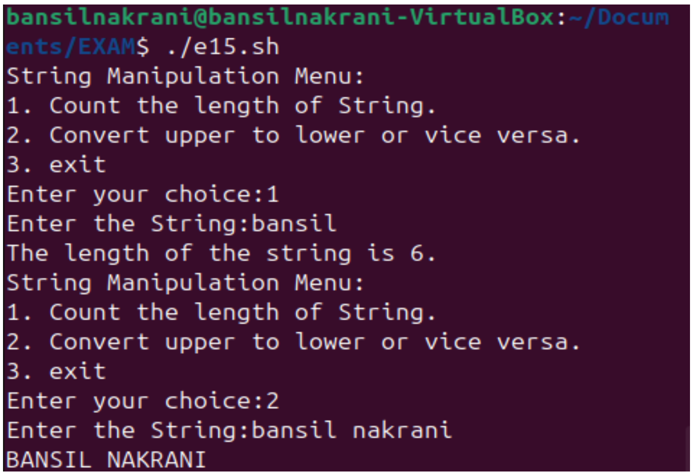
```
bansilnakrani@bansilnakrani-VirtualBox:~/Docum
ents/EXAM$ ./e14.sh
=== CPU temperature ===
sensors command not found. please install.

 === memory usage
               total        used        free
    shared  buff/cache   available
Mem:           7.2Gi       1.5Gi       5.1Gi
      56Mi       878Mi       5.7Gi
Swap:             0B          0B          0B
```

SSP PRACTICAL IMP QUESTION

15.Write a shell script which provide following facilities c. Count the total length of the string. d. Convert upper to lower or vice versa.

```
count_length() {
        echo "The length of the string is ${#1}."
}

convert_case() {
        if [[ $1 == *[A-Z]* ]]; then
          echo "$1" | tr 'A-Z' 'a-z'
        else
          echo "$1" | tr 'a-z' 'A-Z'
        fi
}

while true; do
        echo "String Manipulation Menu:"
        echo "1. Count the length of String."
        echo "2. Convert upper to lower or vice versa."
        echo "3. exit"

        read -p "Enter your choice:" choice


        case $choice in
          1)
          read -p "Enter the String:" str
          count_length "$str"
          ;;
          2)
          read -p "Enter the String:" str
          convert_case "$str"
          ;;
          3)
          exit 0
          ;;
          *)
```

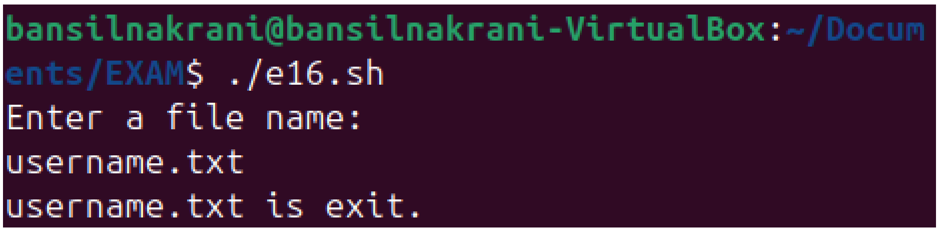
```
            echo "invalid choice. please try again."
            ;;
          esac
done
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/Docum
ents/EXAM$ ./e15.sh
String Manipulation Menu:
1. Count the length of String.
2. Convert upper to lower or vice versa.
3. exit
Enter your choice:1
Enter the String:bansil
The length of the string is 6.
String Manipulation Menu:
1. Count the length of String.
2. Convert upper to lower or vice versa.
3. exit
Enter your choice:2
Enter the String:bansil nakrani
BANSIL NAKRANI
```

16.Write a shell script that takes a filename as an argument and checks if the file exists. If it exists, output "File exists," and if not, output "File does not exist."

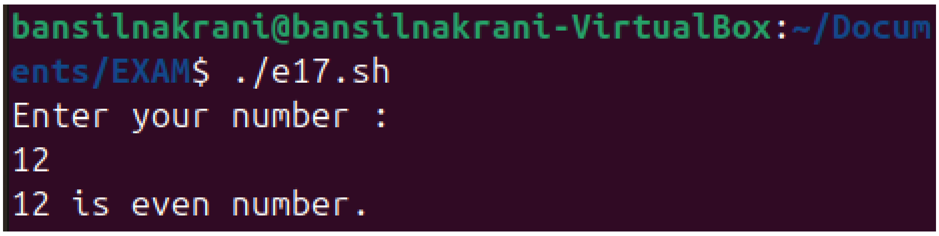
```
echo "Enter a file name:"

read file



if [ -e "$file" ]; then

  echo "$file is exit."

else

  echo "$file does not exit."

fi
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM$ ./e16.sh
Enter a file name:
username.txt
username.txt is exit.
```

17.Write a Shell Bash Script for check if a number input from standard input is odd or even.

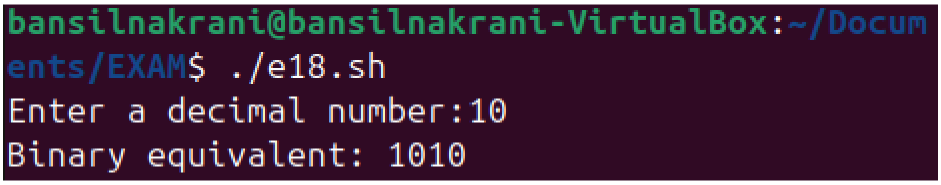
```
echo "Enter your number :"

read num


if (("$num" % 2 == 0)); then

  echo "$num is even number."

else

  echo "$num is odd number."

fi
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM$ ./e17.sh
Enter your number :
12
12 is even number.
```

18.Write a Shell Bash Script for convert Decimal Number to Binary.

```
decimal_to_binary() {
  local decimal=$1
  local binary=""



  while [ $decimal -gt 0 ]; do
      binary=$((decimal % 2))$binary
      decimal=$((decimal / 2))
  done

  echo "$binary"
}

echo -n "Enter a decimal number:"
read decimal

binary=$(decimal_to_binary $decimal)
echo "Binary equivalent: $binary"
```

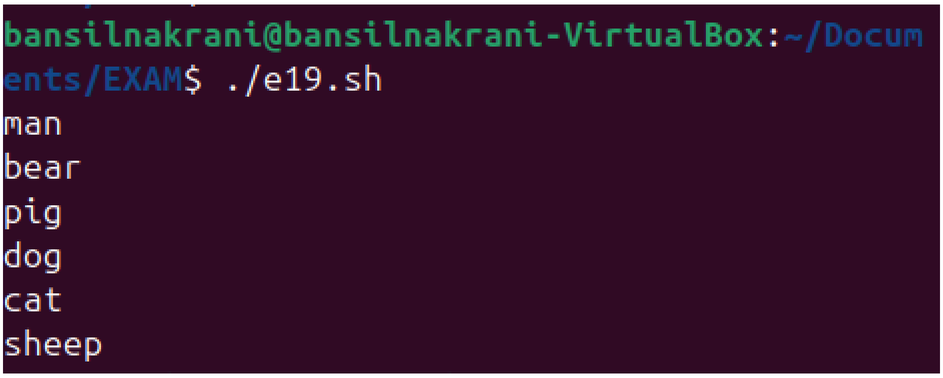


```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM$ ./e18.sh
Enter a decimal number:10
Binary equivalent: 1010
```

SSP PRACTICAL IMP QUESTION

19. Write a shell script that displays “man”,”bear”,”pig”,”dog”,”cat”,and “sheep” on the screen with each appearing on a separate line. Try to do this in as few lines as possible.

```
ANIMALS="man bear pig dog cat sheep"

for ANIMALS in $ANIMALS
do
  echo $ANIMALS
done
```



```
bansilnakrani@bansilnakrani-VirtualBox:~/Documents/EXAM$ ./e19.sh
man
bear
pig
dog
cat
sheep
```