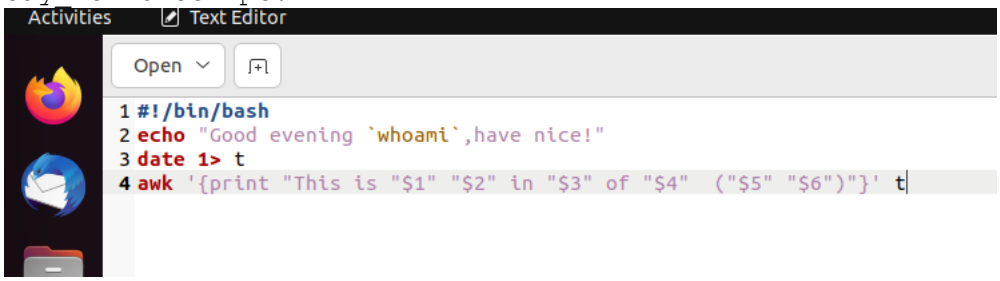
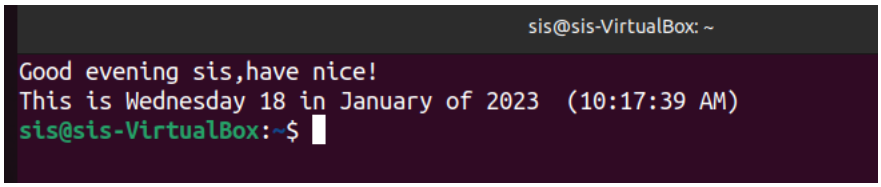
21. Description- Script called say_hello, which will print greetings based on time and to provide date information .
Input- in bashrc file- /home/siddaling/21_say_hello.sh
Output- Good evening siddaling,have nice day!
This is Saturday 09 in January of 2021 (07:58:53 PM)
Input:

```
112  if [ -f /usr/share/bash-completion/bash_completion ]; then
113    . /usr/share/bash-completion/bash_completion
114  elif [ -f /etc/bash_completion ]; then
115    . /etc/bash_completion
116  fi
117 fi
118 /home/sis/say_hello
```

Say_hello script:



Output:



22. Description- Script to convert content of file lower to uppercase and upper to lowercase.
Input- bash 22_upper_lower.sh a.txt
1 – Lower to upper
2 – Upper to lower
please select the option: 1
Output- Before running the script:
Content of the file a.txt- Hello, I am Siddaling Kempasatti,
From Belgaum.

After Running script:
content of the file a.txt- HELLO, I AM SIDDALING KEMPASATTI,
FROM BELGAUM.

Source code:

```
1 #!/bin/bash
2 echo " 1 - Lower to upper"
3 echo " 2 - Upper to lower"
4 read -p "please select the option: " a
5 if [ $a = 1 ]
6 then
7 sed -i 's/[a-z]/\U&/g' $1
8 else
9 cat $1 | tr '[A-Z]' '[a-z]' 1> $1
0 fi
1
```

```
sis@sis-VirtualBox:~$ cat 0< a.txt
Hello, I am Siddaling Kempasatti,
From Belgaum.
sis@sis-VirtualBox:~$ bash 22_upper_lower.sh a.txt
 1 - Lower to upper
 2 - Upper to lower
please select the option: 1
sis@sis-VirtualBox:~$ cat 0< a.txt
HELLO, I AM SIDDALING KEMPASATTI,
FROM BELGAUM.
```

23 . Description- Script to convert content of file lower to uppercase and upper to lowercase.
Input- bash 23_print_fifth_line.sh a.txt
Output- Before running the script:
Content of the file a.txt- Hello, I am Siddaling Kempasatti,
From Belgaum.
PESU, Bangalore.
Electronics and communication,

After Running script:
Output of the script- hELLO, i AM sIDDALING kEMPASATTI,
fROM, bELGAUM.
pesu, bANGALORE.
eLECTRONICS AND COMMUNICATION.

Source code:

```
1 #!/bin/bash
2 echo -n > x.yz
3 while read l
4 do
5   a=`echo $l | wc -c`
6   b=1
7   s=''
8   while [ $b -le $a ]
9   do
10    r=`echo $l | cut -c $b`
11    case $r in
12    [a-z])
13        s=$s`echo $r | tr '[a-z]' '[A-Z]'`
14        ;;
15    [A-Z])
16        s=$s`echo $r | tr '[A-Z]' '[a-z]'`
17        ;;
18    *)
19        s=$s$r
20        ;;
21    esac
22    let b++
23  done
24  echo $s >> x.yz
25 done 0< $1
26 cat x.yz > $1
```

Output:

```
sis@sis-VirtualBox:~$ cat 1> a.txt
Hello, I am Siddaling Kempasatti,
From Belgaum.
PESU, Banglore.
Electronics and communication,
sis@sis-VirtualBox:~$ bash 23_print_fifth_line.sh a.txt
sis@sis-VirtualBox:~$ cat 0< a.txt
hELLO, i AM sIDDALING kEMPASATTI,
fROM bELGAUM.
pesu, bANGLORE.
eLECTRONICS AND COMMUNICATION,
sis@sis-VirtualBox:~$
```

24: Description- Script to use pipes or redirection to create an infinite feedback loop.

Input- bash 24_redirection.sh

Output-
```
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling
Hello Siddaling.................[infinity times]
```

Source code:

Source code:

```
1 #!/bin/bash
2 echo "Hello siddaling" 1> a.txt
3 cat a.txt
4 ./24_redirection.sh
```

Output:

```
sis@sis-VirtualBox:~$ bash 24_redirection.sh
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Help siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
Hello siddaling
```

25. Description-Script to use a recursive function to print each argument passed to the function.
Input-bash 25_recursion.sh 1 2 5 9 4 s jk l 7 5
Output- 1
2
5
9
4
s
jk
l
7
5

Source code:

```
#!/bin/bash
a=$#
n=0
ar=()
for i in $@;
do
  ar[$n]=$i
  let n++
done
n=0
function f()
{
  if [ $n -ge $a ]
  then
    return
  fi
  echo $1
  let n++
  f ${ar[$n]}
}
f ${ar[$n]}
```

Output:

```
sis@sis-VirtualBox:~$ bash 25_recursion.sh 1 2 5 9 s jk 1 7 5
1
2
5
9
s
jk
1
7
5
```

26. Description- Script to determine whether a given file system or mount point is mounted.
Input- bash 26_mounted_fs.sh /dev/sda3
Output-File-system /dev/sda3 is mounted on / and it is having 7% used space with 227859240 KB free

Source code:

```
Open                                    26_mounted_fs.sh
1 #!/bin/bash
2 df > df.txt
3 awk -v a=$1 '{if(a==$1){print "File-system "$1" is mounted on "$6" and it is having "$5" used space with "$4" KB free";exit}}' df.txt
```

Output:

```
sis@sis-VirtualBox:~$ bash 26_mounted_fs.sh /dev/sda3
File-system /dev/sda3 is mounted on / and it is having 52% used space with 11639600 KB free
```

27 . Description- Script that takes any number of directories as command-line arguments and then lists the contents of each of the directories.

Input- bash 27_output_ls.sh siddaling Desktop

Output-

```
siddaling :
/home/siddaling/siddaling :
28_lock_permissions.sh a.c b.c test

/home/siddaling/siddaling/test :
a.txt b.txt c.txt d.txt Emertxe f.sh Siddaling z.sh

Desktop :
/home/siddaling/Desktop :
Emertxe

/home/siddaling/Desktop/Emertxe :
C Documents Linux Systems
```

Source code:

```
1 #!/bin/bash
2 for i in $@
3 do
4   echo $i" : "
5   ls $i
6   echo
7 done
```

Output:

```
sis@sis-VirtualBox:~$ bash 27_output_ls.sh dir1 dir2
dir1 :
a1.txt  a2.txt  a3.txt

dir2 :
a1.txt  a2.txt  a3.txt
```

28. Description- Script to locks file permissions for a particular directory for groups and others

Input- bash 28_lock_permissions.sh test

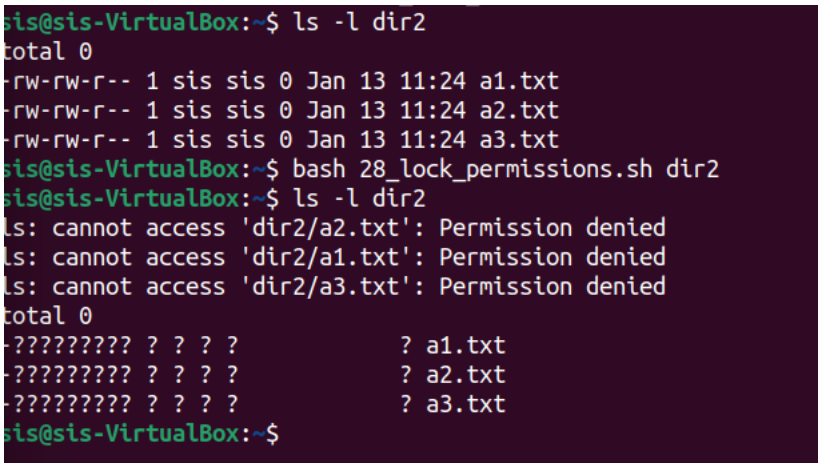
Output-

```
total 8
-rw-rw-r-- 1 siddaling siddaling    0 Jan 17 01:24 a.txt
-rw-rw-r-- 1 siddaling siddaling    0 Jan 17 01:24 b.txt
-rw-rw-r-- 1 siddaling siddaling    0 Jan 17 01:24 c.txt
-rw-rw-r-- 1 siddaling siddaling    0 Jan 17 01:24 d.txt
drwxrwxr-x 2 siddaling siddaling 4096 Jan 17 01:25 Emertxe
-rw-rw-r-- 1 siddaling siddaling    0 Jan 17 01:24 f.sh
drwxrwxr-x 2 siddaling siddaling 4096 Jan 17 01:25 Siddaling
-rw-rw-r-- 1 siddaling siddaling    0 Jan 17 01:24 z.sh
```

```
All files are locked for groups and others
total 8
-rw------- 1 siddaling siddaling    0 Jan 17 01:24 a.txt
-rw------- 1 siddaling siddaling    0 Jan 17 01:24 b.txt
-rw------- 1 siddaling siddaling    0 Jan 17 01:24 c.txt
-rw------- 1 siddaling siddaling    0 Jan 17 01:24 d.txt
drwxrwxr-x 2 siddaling siddaling 4096 Jan 17 01:25 Emertxe
-rw------- 1 siddaling siddaling    0 Jan 17 01:24 f.sh
drwxrwxr-x 2 siddaling siddaling 4096 Jan 17 01:25 Siddaling
-rw------- 1 siddaling siddaling    0 Jan 17 01:24 z.sh
```

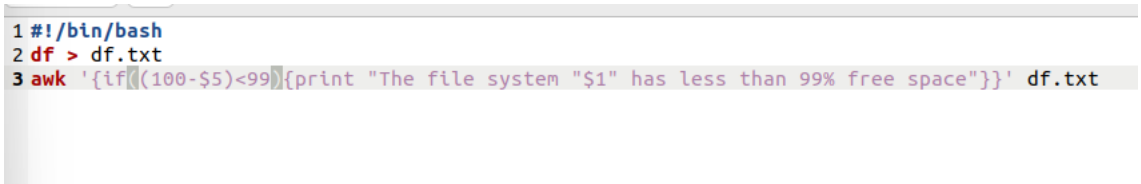
Source code:

Output:

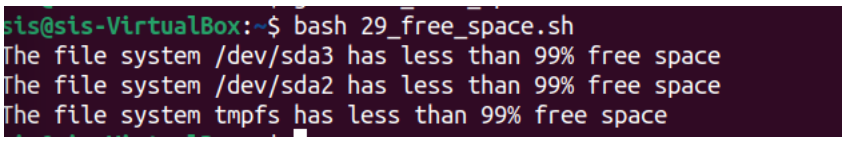



```
29. Description- Script to display the names of any file-system which
have less than 40% free space available
Input- bash 29_free_space.sh
Output-The filesystem /dev/sda5 has less than 40% free space
       The filesystem /dev/sda6 has less than 40% free space
```

Source code:



Output:



30 . Description- Script to count the number of users with user IDs between 500 and 10000 on the system
Input- bash 30_print_user_ids.sh 200 2500
Output-Total count of user ID between 200 to 2500 is : 2
Source code:

```
#!/bin/bash
id -u > u.txt
c=0
for i in `cat u.txt`
do
  if [ $i -lt $2 -a $i -gt $1 ]
  then
      let c++
  fi
done
echo "Total count of user ID between $1 to $2 is : $c"
```

Output:

```
sis@sis-VirtualBox:~$ bash 30_print_user_ids.sh 500 1500
Total count of user ID between 500 to 1500 is : 1
```

31. Description-Script or each directory in the $PATH variable, to display the number of executable files in that directory.
Input- bash 31_executable_path.sh
Output-Current dir: /usr/local/sbin
Current count: 0

Current dir: /usr/local/bin
Current count: 1

Current dir: /usr/sbin
Current count: 387

Current dir: /usr/bin
Current count: 1544

Current dir: /sbin
Current count: 387

Current dir: /bin
Current count: 1544

Current dir: /usr/games
Current count: 0

Current dir: /usr/local/games
Current count: 0

Current dir: /snap/bin
Current count: 5

Total- 3868
Source code:

```
#!/bin/bash
IFS=':'
for i in $PATH
do
  c=0
  echo "Current dir: $i"
  for j in $i/*
  do
    if [ -x $j ]
    then
        let c++
   fi
  done
  echo "Current count: $c"
  echo
done
```

Output:

```
sis@sis-VirtualBox:~$ bash 31_executable_path.sh
Current dir: /usr/local/sbin
Current count: 0

Current dir: /usr/local/bin
Current count: 0

Current dir: /usr/sbin
Current count: 369

Current dir: /usr/bin
Current count: 1360

Current dir: /sbin
Current count: 369

Current dir: /bin
Current count: 1360

Current dir: /usr/games
Current count: 6

Current dir: /usr/local/games
Current count: 0

Current dir: /snap/bin
Current count: 6

Current dir: /snap/bin
Current count: 6
```

32. Description- Script to search a user present in the system.
Input- bash 32_user_present.sh root
Output-User root is present

Source code:

```
#!/bin/bash
getent passwd $1 > /dev/null
if [ $? = 0 ]
then
   echo "User $1 is present"
else
   echo "User $1 is not present"
fi
```

Output

```
sis@sis-VirtualBox:~$ bash 32_user_present.sh root
User root is present
```

33. Description- Script to replace 20% lines in a C file randomly and replace it with the pattern <---DEL--->

Input- bash 33_replace_DEL.sh mycode.c

Output-

```
<-----------Deleted------------>
        int main()
        {
                int num, rem, reverse_num, temp, start, end;
                printf("Enter the lower limit: ");
                scanf("%d", &start);
                printf("Enter the upper limit: ");
        <-----------Deleted------------>
                printf("Palindrome numbers between %d and %d are: ",
start, end);
                for (num = start; num <= end; num++)
        <-----------Deleted------------>
                        temp = num;
                        reverse_num = 0;
                        while (temp)
                        {
                                rem = temp % 10;
                                temp = temp / 10;
                                reverse_num = reverse_num *10 + rem;
                        }
                        if (num == reverse_num)
                                printf("%d ", num);
                }
        <-----------Deleted------------>
        }
```

Source code:

```
#!/bin/bash
ar=()
n=`cat $1 | wc -l`
let nd=n/5
for((i=0;i<nd;i++))
do
   t=0
   while [ $t = 0 ]
   do
      let r=(${RANDOM}%n)+1
      y=0
      for j in $ar[@]
      do
         if [ $j = $r ]
         then
             y=1
             break
         fi
      done
      if [ $y = 0 ]
      then
          t=1
      fi
   done
   ar[$i]=$r
   sed -i $r's/.*/<--------------------DEL------------------------>/' $1
done
cat $1
```

Output:

```
sis@sis-VirtualBox:~$ bash 33_replace_DEL.sh file.c
<--------------------DEL------------------------>
int main(){
        int i=0;
        printf("ddr");
}
```

34. Description- Script to to calculate the BMI.

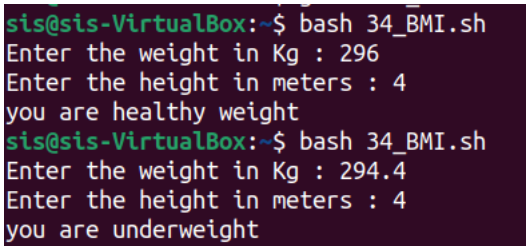
Input- bash 34_BMI.sh
Enter the weight in Kg :48.2
Enter the height in meters :1.4

Output- The BMI is 25.3
You are overweight

Source code:

```
#!/bin/bash
read -p "Enter the weight in Kg : " w
read -p "Enter the height in meters : " h
bmi=`echo $w/$h^2 | bc`
b=`echo $bmi+0.5 | bc`
c=`echo "$b*$h^2" | bc`
if [ `echo "$bmi > 29.9" | bc` = 1 ];then
    echo "you are obese"
elif [ `echo "$bmi > 24.9" | bc` = 1 ];then
    echo "you are overweight"
elif [ `echo "$bmi > 18.4" | bc` = 1 ];then
    echo "you are healthy weight"
elif [ `echo "$c <= $w" | bc` = 1 ] ;then
    echo "you are healthy weight"
else
   echo "you are underweight"
fi
```

Output:


sis@sis-VirtualBox:~$ bash 34_BMI.sh
Enter the weight in Kg : 296
Enter the height in meters : 4
you are healthy weight
sis@sis-VirtualBox:~$ bash 34_BMI.sh
Enter the weight in Kg : 294.4
Enter the height in meters : 4
you are underweight