**Practical : 1**

1.



2.



3.



4.



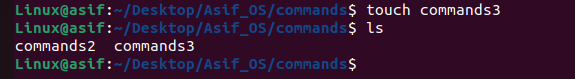
5.



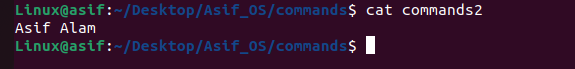
6.



7.



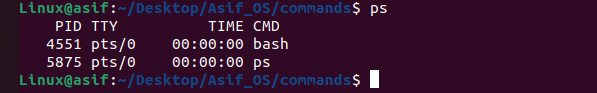
8.



9.



10.

11.

Practical 1: Demonstrate different commands of Linux (min. 15 commands).

1. cd – change directory

cd directory\_name #syntax

This command is used to switch between different directories.

2. ls – list

ls #syntax

This command is used to display all the files or directory in current directory.

3. mkdir – make directory

mkdir folder\_name #syntax

This command is used to make a new directory.

4. rmdir – remove directory

rmdir folder\_name #syntax

This command is used to delete whole directory (if directory is empty).

5. cp – copy

cp source\_file destination\_file #syntax

This command is used to copy one file content to another one.

6. mv – move

mv source\_folder destination\_folder #syntax

This command is used to move any folder to another directory.

7. rm – remove

rm file\_name #syntax

This command is used to remove/delete the file.

8. touch – create file

torch file\_name #syntax

This command is used to create a new file in current directory.

9. cat – content display

cat file\_name #syntax

This command is used to display the content of a file.

10. man – manual

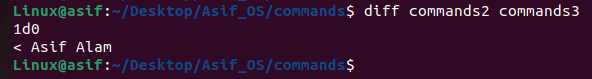
man command\_name #syntax

This command is used to display all the functionality of any particular command.

11. ps – process

ps #syntax

This command is used to display all the process currently running.



12.



13.

A screenshot of a computer

Description automatically generated

14.

15.

12. diff – difference

diff filename1 filename2 #syntax

This command is used to compare the difference between two files.

13. sort – sorting

sort filename #syntax

This command is used to sort the file in alphabetical order.

14. top

Top #syntax

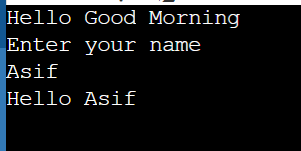
This command is used as a CLI version of task manager.

15. passwd – password

Passwd #syntax

This command is used to change the password.

**Practical 2**

****

**2.**

**Practical 2: Demonstrate commands of Unix.**

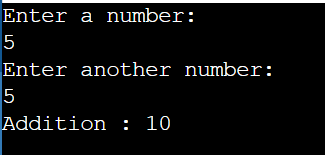
echo "Hello Good Morning";

echo "Enter your name";

read name

echo "Hello $name";

**Practical 3**

**a.**

A screenshot of a computer

Description automatically generated**b.**

**Practical 3**

**a: Write a shell script to display addition of given two numbers.**

**Code:**

echo "Enter a number: "

read a

echo "Enter another number: "

read b

res=`expr $a + $b`

echo "Addition: $res"

**b: Write a shell script to display multiplication table of given number.**

**Code:**

echo "Enter a number: "

read num

i=1

while [ $i -le 10 ]

do

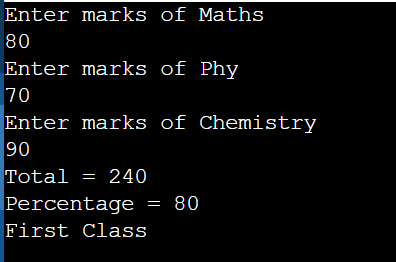
res=`expr $num \\* $i`

echo "$num \* $i = $res"

((++i))

done

**Practical 4**

**4.**

**Practical 4: Write a shell script to generate marksheet of a student. Take 3 subjects, calculate and display total marks, percentage and class obtained by the student.**

**Code:**

echo "Enter marks of Maths"

read Maths

echo "Enter marks of Phy"

read Phy

echo "Enter marks of Chemistry"

read chem

total=expr $Maths + $Phy + $chem

perc=`expr $total / 3`

echo "Total = $total"

echo "Percentage = $perc"

if [ $perc -ge 70 ]

then

echo "First Class"

elif [ $perc -ge 60 ]

then

echo "Second Class"

elif [ $perc -ge 50 ]

then

echo "Third Class"

elif [ $perc -lh 50 ]

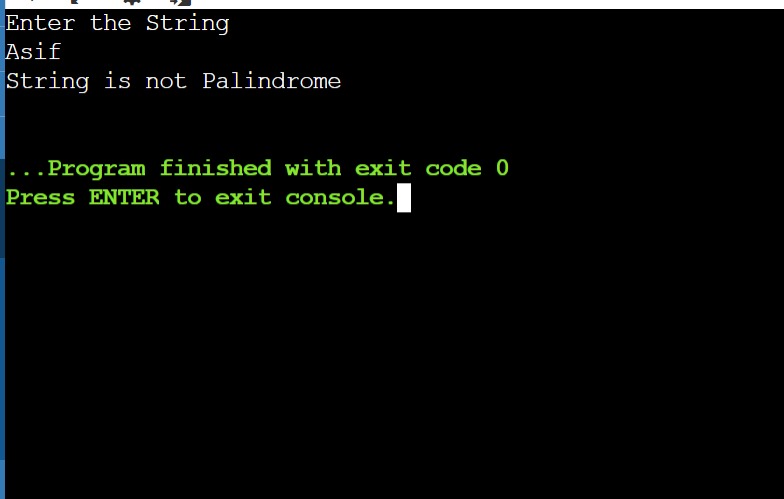
then

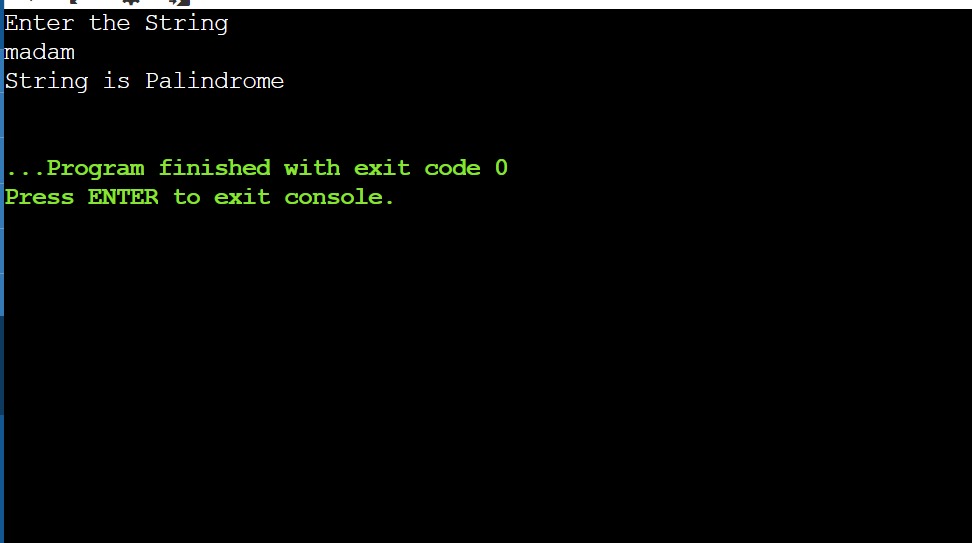
echo "Fail"

fi

**Practical 5**

**5.**





**5. Write a shell script to check entered string is palindrome.**

echo "Enter the String"

# Taking the input.

read inp

# Taking Extra variable for reversing the string.

rev=""

#Finding the length of string.

len=${#inp}

#Traversing the string.

for((i=$len-1;i>=0;i--))

do

# #Storing reverse value

rev="$rev${inp:$i:1}"

done

#Checking if String is Palindrome or not .

if [ $inp == $rev ]

then

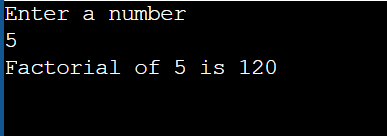
echo "String is Palindrome"

else

echo "String is not Palindrome"

fi

**Practical 6**

**6.**

**6. Write a shell script to find factorial of given number n.**

echo "Enter a number"

read number

fact=1

for (( i = 1; i <= number; i++ )) do

fact=`expr $i \\* $fact`

done

echo "Factorial of $number is $fact"

**Practical 7**

**A black screen with white text

Description automatically generatedOutput:**

A black screen with white text

Description automatically generated

**7. Write a shell script which will accept a number b and display whether it is prime numbers or not.**

echo "Enter a number"

read n

count=0

for (( i = 2; i < $n; i++ ))

do

x=`expr $n \% $i`

if [ $x == 0 ]

then

count=1

break

fi

done

if [ $count == 1 ]

then

echo "$n is not prime"

else

echo "$n is a prime"

fi

**Practical 8**

**A black screen with white text

Description automatically generatedOutput:**

**A screenshot of a computer

Description automatically generated**

**8. Write a shell script which will generate first n Fibonacci numbers like: 1,1,2,3,5,8,13,21….**

echo "Enter a number"

read n

a=1

b=1

z=0

echo "$z"

echo "$a"

echo "$b"

for (( i = 1; i <=n-3; i++ )); do

c=`expr $b \+ $a`

temp=$a

a=$b

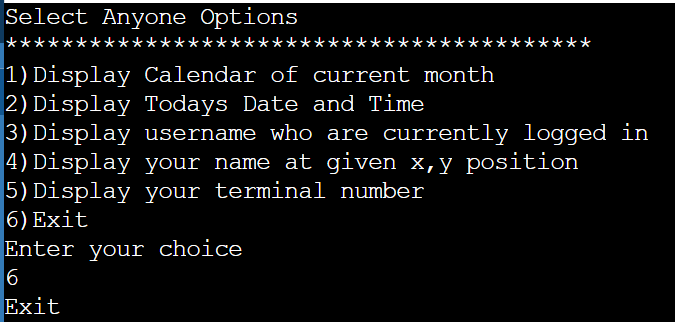
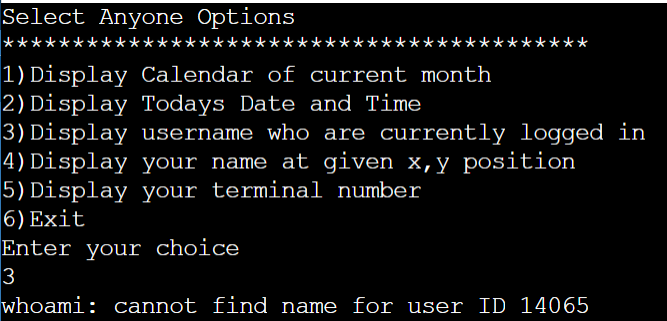
b=`expr $temp + $b`

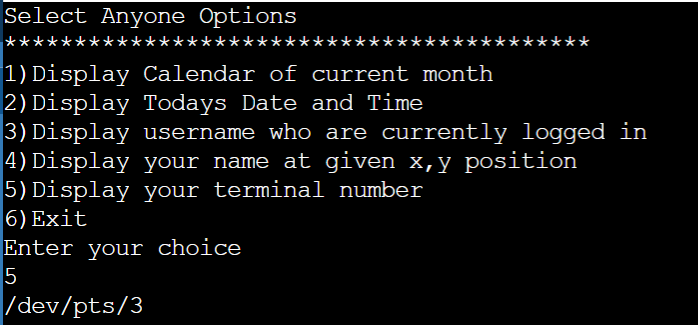
echo "$c "

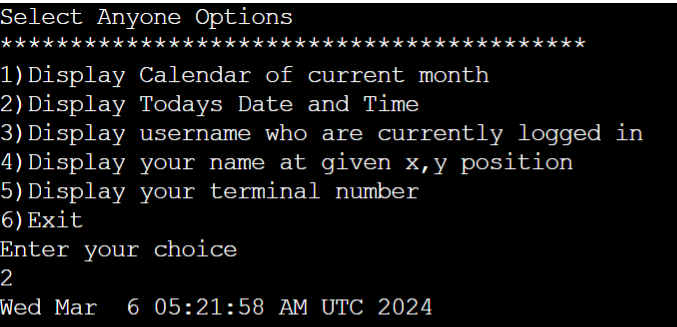
done

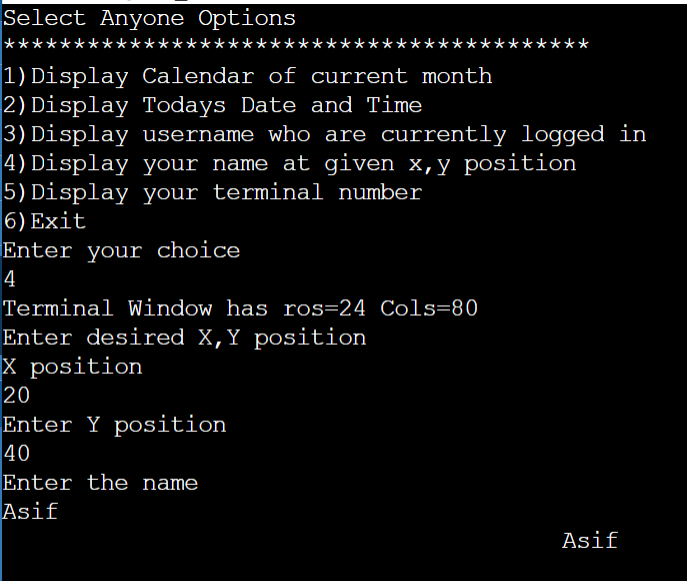
**Practical 9**

**Output:**

****

****

****

****

**9. Write a menu driven shell script which will print the following menu and execute the given task.**

**a. Display calendar of current month**

**b. Display today’s date and time**

**c. Display usernames those are currently logged in the system**

**d. Display your name at given x, y position**

**e. Display your terminal number**.

echo "Select Anyone Options";

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "1)Display Calendar of current month"

echo "2)Display Todays Date and Time"

echo "3)Display username who are currently logged in"

echo "4)Display your name at given x,y position"

echo "5)Display your terminal number"

echo "6)Exit"

echo "Enter your choice"

read ch

case $ch in

1)cal ;;

2)date;;

3)whoami;;

4)row=$(tput lines)

col=$(tput cols)

echo "Terminal Window has ros=$row Cols=$col"

echo "Enter desired X,Y position"

echo "X position"

read x

echo "Enter Y position"

read y

echo "Enter the name"

read name

tput cup $x $y

echo "$name";;

5)tty;;

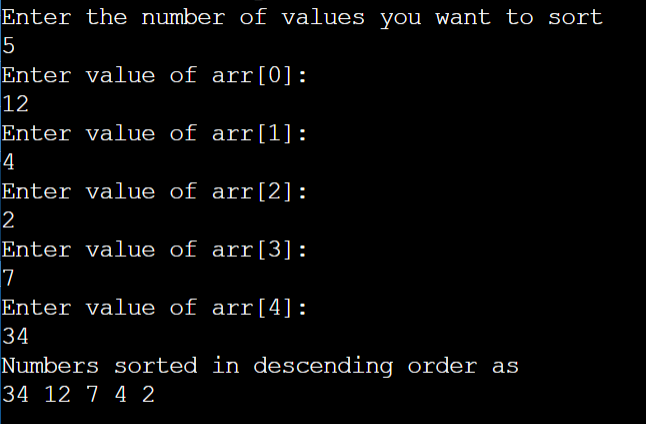
6)echo "Exit";;

\*)echo "Enter valid choice"

Esac

**Practical 10**

**Output:**

****

**10. Write a shell script to read n numbers as command arguments and sort them in descending order.**

echo "Enter the number of values you want to sort "

read n

for((i=0; i<$n; i++))

do

echo "Enter value of arr[$i]: "

read arr[$i]

done

#sorting code

for((i=0; i<$n; i++))

do

for((j=0; j<n-i-1; j++))

do

if [ ${arr[j]} -lt ${arr[$((j+1))]} ]

then

#swapping

temp=${arr[j]}

arr[$j]=${arr[$((j+1))]}

arr[$((j+1))]=$temp

fi

done

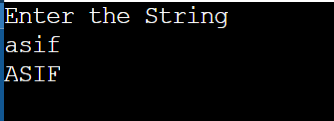
done

echo "Numbers sorted in descending order as"

echo ${arr[\*]}

**Practical 11**

**Output:**

****

**11. Write a program using function, which convert each word in a given text into capital.**

echo "Enter the String"

# Read user input into the variable 'str'

read str

# Convert the input string to uppercase using awk

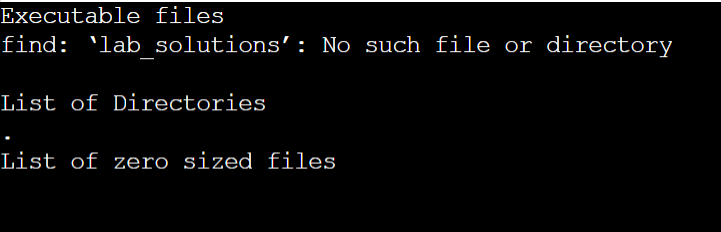
a=$(echo "$str" | awk '{print toupper($0)}')

# Output the uppercase string

echo $a

**Practical 12**

**Output:**

****

**12. Write a shell script to display all Executable Files, Directories And Zero sized files from current directory.**

echo 'Executable files'

files=$(find lab\_solutions -executable -type f)

echo $files

echo 'List of Directories'

dir=$(ls -d )

echo $dir

echo 'List of zero sized files'

zero=$(find -size 0)

echo $zero