**Program No.1** PROGRAM 1 Date:- 5/8/2024

**AIM**:- Write a scripts to read any 2 floating values and find the sum,difference,quotient, and remainder

**SOURCE CODE**

```
#!/bin/bash

echo Program to show Arithmetic operation on floating point numbers

echo Enter the first number

read a

echo Enter the second number

read b

sum1=`expr "$a + $b" | bc`

product=`expr "$a * $b" | bc`

div=`expr "$a / $b" | bc`

mod=`expr "$a % $b" | bc`

diff=`expr "$a - $b" | bc`

echo The sum is : $sum1

echo The difference is : $diff

echo The product is : $product

echo The division result is : $div

echo The remainder is : $mod
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$ bash 1.sh
Program to show Arithmetic operation on floating point numbers
Enter the first number:
22
enter the second number:
2
Sum :24
Diff:20
Pro:44
Div:11
Mod:0
macfast@macfast-V530S-07ICR:~/mcalab$
```

**PROGRAM 2** Date:-5/8/2024

**AIM:-** Write a script to read the length and breadth of a rectangle and radius of a circle and calculate the area and perimeter of the rectangle and circumference of the circle

**SOURCE CODE**

```
#!/bin/bash

echo The Area, Perimeter and Circufernce Calculator

echo enter the length of the rectange

read length

echo enter the breadth of the rectangle

read width

area=`expr "$length * $width" | bc`

perim=`expr "2 * ($length + $width)" | bc`

echo area is $area

echo perimeter is $perim

echo Enter the radious

read rad

vol=`expr "2 * 3.1415 * $rad" |bc`

cir=`expr "3.1415 * $rad * $rad" |bc`

echo the area of circle is $vol

echo the circumference is $cir
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$ bash 2.sh
The Area, Perimeter and Circumference Calculator
Enter the length of the rectangle:
2
Enter the breadth of the rectangle:
3
Area of rectangle is 6
Perimeter of rectangle is 10
enter the radius of circle:
2
Area of circle: 12.5660
Circumference of circle: 12.5660
macfast@macfast-V530S-07ICR:~/mcalab$
```

**PROGRAM 3** Date:- 8/8/2024

**AIM:-** Write a shell program to find

a) Sum of digits of a number
b) Reverse of the number
c) Determine whether the given number is a palindrome or not

**SOURCE CODE**

```
#!/bin/bash
echo The program to find the sum of digits, Palindrome, and Reverse
echo enter the number to reverse, sum of digits, palindrome
read num
dsum=0
rev=0
num1=$num
while ((num > 0))
do
check=$((num % 10))
((dsum = check + dsum))
((num = num / 10))
((rev = rev * 10 + check))
done
echo The Sum is: $dsum
echo The Reverse is: $rev
if [ $rev -eq $num1 ]
then
echo the number is palindrome
else
echo the number is not palindrome
fi
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$ bash 3.sh
The program to find the sum of digits, Palindrome, and Reverse
Enter the number to reverse, sum of digits, palindrome
121
The sum is:4
The Reverse is:121
The number is palindrome
macfast@macfast-V530S-07ICR:~/mcalab$ bash 3.sh
The program to find the sum of digits, Palindrome, and Reverse
Enter the number to reverse, sum of digits, palindrome
213
The sum is:6
The Reverse is:312
The number is not palindrome
macfast@macfast-V530S-07ICR:~/mcalab$
```

**PROGRAM 4** Date:- 8/8/2024

**AIM:-** Write a shell script to display the digits which are in odd positions in a given integer

**SOURCE CODE**

```
#!/bin/bash
n=0
echo The program to find the number at odd position
read -p "Enter a number : " n
echo $n
l=${#n}
i=1
echo The odd position numbers are
while [ $i -le $l ]
do
d=$(echo $n | cut -c $i)
echo -n $d " "
((i= i + 2))
done
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$ bash 4.sh
The program to find the number at odd position
Enter a number:
1234567890
The odd position numbers are
1
3
5
7
9
macfast@macfast-V530S-07ICR:~/mcalab$
```

**PROGRAM 5** Date:- 12/8/2024

**AIM:-** Write a script to read the basic salary of n employees and calculate the gross salary

If BP<15000, DA=30% of BP, HRA =Rs 500. TA = 10% of BP.

If BP >=5000, DA=50% of BP, HRA=15%, TA=1000.

**SOURCE CODE**

```
#!/bin/bash
echo The program to find Gross Salary
echo Enter total no of employees
read n
count=1
while [ $count -le $n ]
do
echo enter the basic salary
read bsalary
if [ $bsalary -lt 15000 ]
then
gsal=`expr "$bsalary + ($bsalary * .3) + 500 +
($bsalary * .1)" | bc`
echo the gross salary is $gsal
else
gsal=`expr "$bsalary + ($bsalary * .5) + 1000
+ ($bsalary * .15)" | bc`
echo the gross salary is $gsal
fi
((count = count + 1))
done
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$ bash 5.sh
The program to find Gross Salary
Enter the total no of employees
2
enter the basic salary:
10000
Gross salary is 14500.00
enter the basic salary:
20000
Gross salary is 34000.00
macfast@macfast-V530S-07ICR:~/mcalab$
```

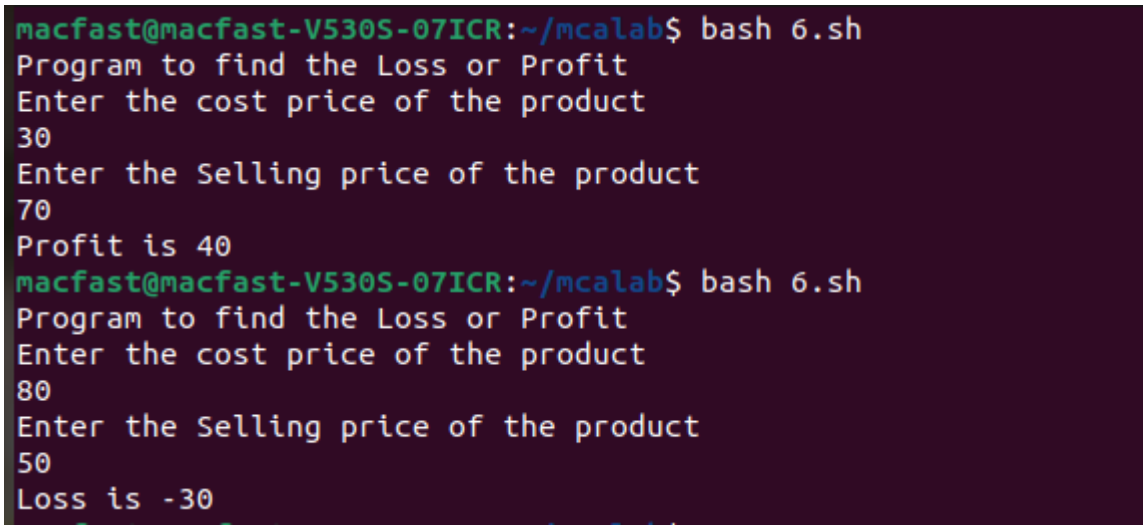
**PROGRAM 6** Date:- 12/8/2024

**AIM:-** Write a script to read the cost and selling price of an item and to decide how much loss or profit has incurred by the seller

**SOURCE CODE**

```
#!/bin/bash
echo Program to find the Loss or profit
echo Enter the cost price of the product
read cost
echo Enter the Selling price of the product
read sell
if [ $cost -gt $sell ]
then
loss=`expr "$sell - $cost" | bc`
echo loss is $loss
else
profit=`expr "$sell - $cost" | bc`
echo profit is $profit
fi
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 6.sh
Program to find the Loss or Profit
Enter the cost price of the product
30
Enter the Selling price of the product
70
Profit is 40
macfast@macfast-V530S-07ICR:~/mcalab$ bash 6.sh
Program to find the Loss or Profit
Enter the cost price of the product
80
Enter the Selling price of the product
50
Loss is -30

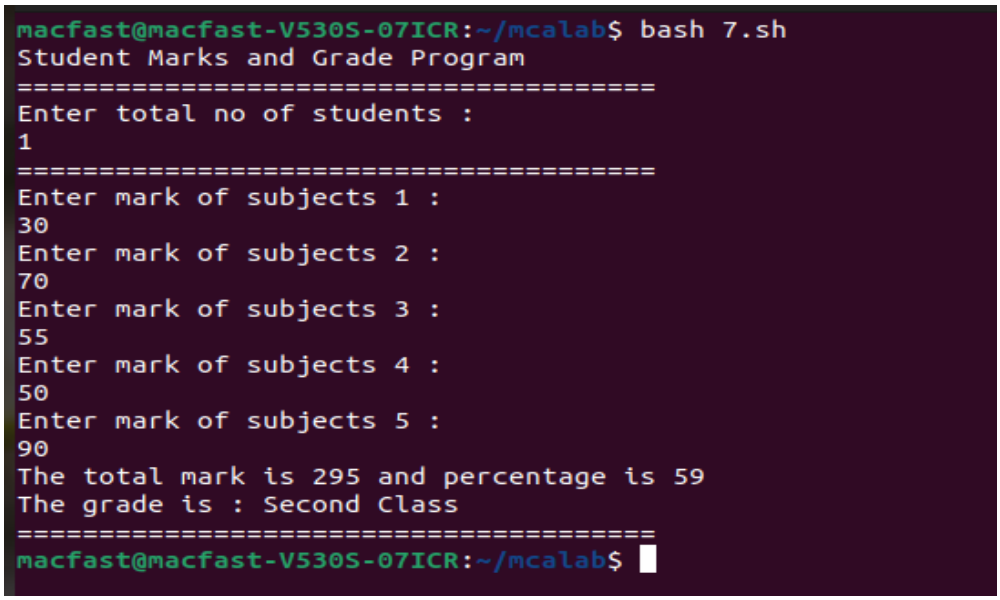
**PROGRAM 7** Date:- 19/8/2024

**AIM:-** Write a script to read 5 marks of n students. Find the total and return distinction if the total percentage >= 80. [Distinction] if total % is >=60 and <80 [first class].if total % is >=50 and <60 [second class] else print failed [<50].

**SOURCE CODE**

```
#!/bin/bash

echo Student Marks and Grade Program

echo ====================================

echo "enter Total no of students :"

read n

count=1

while [ $count -le $n ]

do
total=0
for((i=1; i <= 5;i++))

do

echo "enter marks of subjects $i : "

read marks

total=$(( $total + $marks))

done

percent=`expr "$total * 100 / 500" |bc`

echo The Total Marks is $total and Percentage is $percent
if [ $percent -ge 80 ]
then
echo The Grade is: Distinction
elif test $percent -ge 60

then

echo The Grade is: First Class

elif test $percent -ge 50

then
echo The Grade is: Second Class
else
echo Failed
fi
((count= count + 1))
echo ====================================
done
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 7.sh
Student Marks and Grade Program
Enter total no of students :
1
Enter mark of subjects 1 :
30
Enter mark of subjects 2 :
70
Enter mark of subjects 3 :
55
Enter mark of subjects 4 :
50
Enter mark of subjects 5 :
90
The total mark is 295 and percentage is 59
The grade is : Second Class
macfast@macfast-V530S-07ICR:~/mcalab$

**PROGRAM 8** Date:-19/8/2024

**AIM**:- Write a script to read a character and to display if it is lowercase, uppercase, digit or special character or not a character.

**SOURCE CODE**

```
#!/bin/bash

echo The program to find an entered is upper / lower / alpha numeric

echo Enter the character

read ch

if [[ "$ch" =~ [ABCDEFGHIJKLMNOPQRSTUVWXYZ] ]]

then

echo The character is UPPERCASE

elif [[ "$ch" =~ [a-z] ]]

then

echo The character is Lowercase

elif [[ "$ch" =~ [0-9] ]]

then

echo The character is a Digit

elif [[ "$ch" =~ ['!@#\$%^\&*()_-+'] ]]

then

echo it is a Special Character

else

echo Nothing matching....

fi
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$ bash 8.sh
the program to find an entered is upper / lower/ alpha numeric
Enter the character
a
The character is LOWERCASE
macfast@macfast-V530S-07ICR:~/mcalab$ bash 8.sh
the program to find an entered is upper / lower/ alpha numeric
Enter the character
A
The character is UPPERCASE
macfast@macfast-V530S-07ICR:~/mcalab$ bash 8.sh
the program to find an entered is upper / lower/ alpha numeric
Enter the character
2
The character is DIGIT
macfast@macfast-V530S-07ICR:~/mcalab$
```

**PROGRAM 9** Date:-22/8/2024

**AIM:-** Write a script to prepare a multiplication table of a given number to any order.

**SOURCE CODE**

```
#!/bin/bash
echo The multiplication table
echo Enter the number to find the multipliation table
read n
echo Enter the Range upto where the table required
read r
i=0
while [ $i -le $r ]
do
echo " $n x $i = `expr $n \* $i` "
i=`expr $i + 1`
done
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 9.sh
Enter the number to find the multiplication table
5
Enter the range upto where the table required
10
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

**PROGRAM 10** Date:- 22/8/2024

**AIM:-** Write a script to find the value of one number raised to the power(a)^n

**SOURCE CODE**

```
#!/bin/bash
echo Program to find the power of number
echo "Input number"
read a
echo "Input power"
read b
counter=0
ans=1
while [ $b -ne $counter ]
do
ans=`expr $ans \* $a`
counter=`expr $counter + 1`
done
echo "$a power of $b is $ans"
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$ bash 10.sh
Program to find the power of number
Input number
5
input power
2
5 power of 2 is 25
macfast@macfast-V530S-07ICR:~/mcalab$
```

**PROGRAM 11** Date:- 29/8/2024

**AIM:-** Write a script to print all prime numbers from 1 to n

**SOURCE CODE**

```
#!/bin/bash
echo Program to find prime between 1 to n
echo enter the upper limit
read upper
flag=0
number=2
echo The prime numbers are:
while [ $number -le $upper ]
do
flag=0
for((i=2; i<=$number/2; i++))
do
num=$((number % i))
if [ $num -eq 0 ]
then
flag=1
break
fi
done
if [ $flag -eq 0 ]
then
echo -n $number " "
fi
((number = number + 1))
done
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$ bash 11.sh
Program to find prime between 1 to n
Enter the upper limit
20
The prime numbers are:
2 3 5 7 11 13 17 19 macfast@macfast-V530S-07ICR:~/mcalab$
```

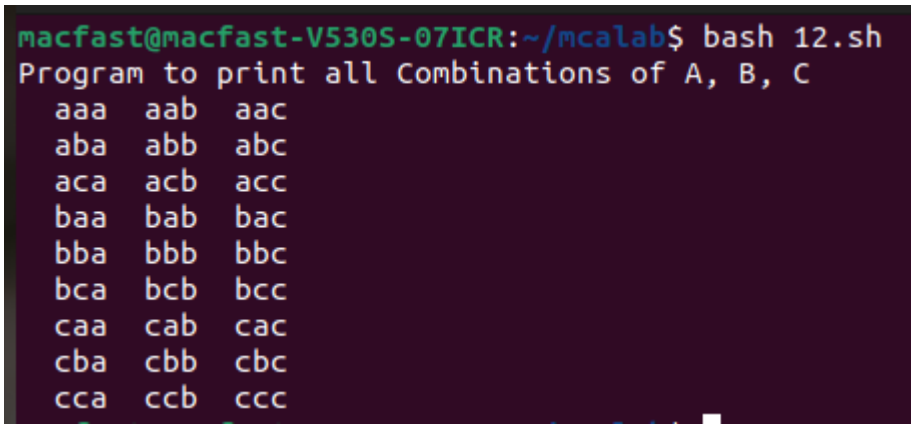
**PROGRAM 12** Date:- 2/9/2024

**AIM:-** Write a script to generate all combinations of a, b and c.

**SOURCE CODE**

```
#!/bin/bash
echo Program to print all Combinations of A, B, C
for c1 in {a..c}
do
for c2 in {a..c}
do
for c3 in {a..c}
do
echo -n " " "$c1$c2$c3"
done
echo
done
done
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 12.sh
Program to print all Combinations of A, B, C
aaa aab aac
aba abb abc
aca acb acc
baa bab bac
bba bbb bbc
bca bcb bcc
caa cab cac
cba cbb cbc
cca ccb ccc

**PROGRAM 13** Date:- 2/9/2024

**AIM:-** Write a shell script to sum up the following series

1/1! + 2/2! + 3/3! + ….

**SOURCE CODE**

```
#!/bin/bash
function fact() {
local n=$1
if [[ $n -eq 0 ]]; then
echo 1
else
echo $((n*$(fact $n-1)))
fi
}
#Factorial function
echo program to find the sum of series
echo enter the upper limit
read n
sum=0
for ((i=1;i<=n;i++))
do
sum=`expr "$sum + ($i / $(fact $i))" |bc`
done
echo “Sum n series is $sum”
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$ bash 13.sh
program to find the sum of series
enter the upper limit
1
“Sum n series is 1”
macfast@macfast-V530S-07ICR:~/mcalab$
```

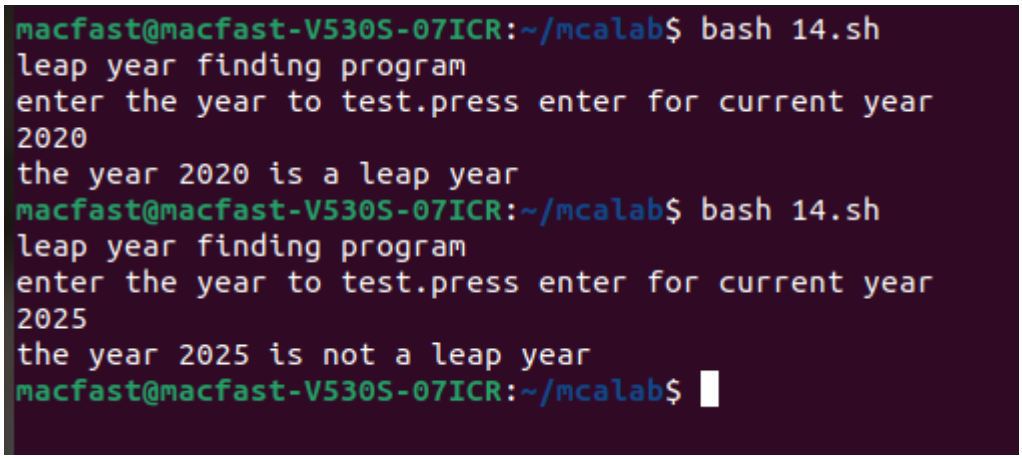
**PROGRAM 14** Date:- 5/9/2024

**AIM:-** Write a script to read a year and to decide whether it is a leap year or not. If no year is supplied then the current year is assumed.

**SOURCE CODE**

```
#!/bin/bash

echo leap year finding Program

echo enter the year to test. press enter for current year

read year #if null entry will assign current year

if [ -z "$year" ]

then

year=$(date +"%Y")

fi

check=`echo $year % 4 | bc`

if [ $check -eq 0 ]

then

if [ `echo $year % 100 |bc ` -eq 0 ]

then

if [ `echo $year % 400 |bc` -eq 0 ]

then

echo the year $year is a leap year

else

echo the year $year is not a leap year

fi

else

echo the year $year is a leap year

fi

else

echo the year $year is not a leap year

fi
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 14.sh
leap year finding program
enter the year to test.press enter for current year
2020
the year 2020 is a leap year
macfast@macfast-V530S-07ICR:~/mcalab$ bash 14.sh
leap year finding program
enter the year to test.press enter for current year
2025
the year 2025 is not a leap year
macfast@macfast-V530S-07ICR:~/mcalab$

**PROGRAM 15** Date:- 5/9/2024

**AIM:-** Shell script to perform operations like display, list, make directory and copy, rename, delete file.

**SOURCE CODE**

```
#!/bin/bash

while :

do

clear

echo The program to List, Display, Make Dir, Copy and Rename File

printf "\nSELECT YOUR OPTION\n"

echo "1. List Directory"

echo "2. Create Directory"

echo "3. Copy Files"

echo "4. Display Files"

echo "5. Rename a File"

echo "6. Delete a file"

echo "7. Exit from program"

echo "Enter your menu choice [1-7]: "

# Running a forever loop using while statement

# This loop will run until select the exit option.

# User will be asked to select option again and again

# reading choice

read choice

# case statement is used to compare one value with the multiple

cases.

case $choice in

# Pattern 1

1) ls

read -p "Press any key...: " ;;

# Pattern 2

2) echo "Enter The Directory Name"

read dirName

mkdir $dirName;
```

```
echo Directory Created...

read -p "Press any key...: " ;;

# Pattern 3

3) read -p "Enter name of the file to be copied :" fname

read -p "Enter the name of the new file :" newname

cp $fname $newname

echo File copied...

read -p "Press any key...: " ;;

# Pattern 4
4) echo Enter the file name to be read
read fname

cat $fname

read -p "Press any key...: " ;;

# Pattern 5

5) echo Enter the file name

read fname

echo Enter the new name to be renamed

read newname

mv $fname $newname

echo file renamed...

read -p "Press any key...: " ;;

# Pattern 6
6) echo Enter the file name
read fname

rm $fname

echo file deleted...

read -p "Press any key...: " ;;

# Pattern 7

7) echo "Quitting ..."

exit;;

# Default Pattern

*) echo "invalid option";;

esac
done
```

## OUTPUT

```
The program to List, Display, Make Dir, Copy and Rename File

SELECT YOUR OPTION
1. List Directory
2. Create Directory
3. Copy Files
4. Display Files
5. Rename a File
6. Delete a file
7. Exit from program
Enter your menu choice [1-7]:
1
14.sh: line 21: cases.: command not found
 14.sh              19_allfile.sh          27_mca.sh             '=5000,'           bubble_23.sh         fact_24.sh    gross_sal.sh   odd_pos.sh
 16_menu.sh         20_samefile.sh         28_file_dir_check.sh   6_sell.sh         comb.sh              fact_sum.sh   multi.sh       power.sh
 17_terminal.sh     21_fileexit.sh         29.sh                  7_grade.sh        count_line_word.sh   file1.txt     multi.sh.txt   prime.sh
 18_fileappend.sh   22_student_manage.sh   30_file_copy.sh        8_char_check.sh   etc                  file3.txt     myfile.txt     stud_data.dt
Press any key...:
```

```
The program to List, Display, Make Dir, Copy and Rename File

SELECT YOUR OPTION
1. List Directory
2. Create Directory
3. Copy Files
4. Display Files
5. Rename a File
6. Delete a file
7. Exit from program
Enter your menu choice [1-7]:
2
14.sh: line 21: cases.: command not found
Enter The Directory Name
mydir
Directory Created...
Press any key...:
```

```
The program to List, Display, Make Dir, Copy and Rename File

SELECT YOUR OPTION
1. List Directory
2. Create Directory
3. Copy Files
4. Display Files
5. Rename a File
6. Delete a file
7. Exit from program
Enter your menu choice [1-7]:
3
Enter name of the file to be copied :myfile.txt
Enter the name of the new file :hello.txt
File copied...
Press any key...:
```

```
The program to List, Display, Make Dir, Copy and Rename File

SELECT YOUR OPTION
1. List Directory
2. Create Directory
3. Copy Files
4. Display Files
5. Rename a File
6. Delete a file
7. Exit from program
Enter your menu choice [1-7]:
4
Enter the file name to be read
myfile.txt
mca is 2 yr program
mca is the master of the bca
mca is good path for higher education
mca is short for master of computer appliocation
hello dear
Press any key...:
```

```
The program to List, Display, Make Dir, Copy and Rename File

SELECT YOUR OPTION
1. List Directory
2. Create Directory
3. Copy Files
4. Display Files
5. Rename a File
6. Delete a file
7. Exit from program
Enter your menu choice [1-7]:
5
Enter the file name
myfile.txt
Enter the new name to be renamed
urfile.txt
file renamed...
Press any key...:
```

```
The program to List, Display, Make Dir, Copy and Rename File

SELECT YOUR OPTION
1. List Directory
2. Create Directory
3. Copy Files
4. Display Files
5. Rename a File
6. Delete a file
7. Exit from program
Enter your menu choice [1-7]:
6
Enter the file name
urfile.txt
file deleted...
Press any key...:
```

```
The program to List, Display, Make Dir, Copy and Rename File

SELECT YOUR OPTION
1. List Directory
2. Create Directory
3. Copy Files
4. Display Files
5. Rename a File
6. Delete a file
7. Exit from program
Enter your menu choice [1-7]:
7
Quitting ...
```

**PROGRAM 16** Date:-23/9/2024

**AIM**:- Write a menu driven program to display the following options.

• Contents of /etc/passwd

• List of output of ‘who’

• Present working directory

• Exit

**SOURCE CODE**

```
#!/bin/bash

while :

do

clear

echo Write a menu driven program to display the following options

printf "\nSELECT YOUR OPTION\n"

echo "1. Contents of /etc/passwd"
echo "2. List of output of ‘who’"
echo "3. Present working directory"
echo "4. Exit from program"
echo "Enter your menu choice [1-7]: "
# reading choice
read choice
# case statement is used to compare one value with the multiple
cases.
case $choice in

# Pattern 1
1) cat etc/passwd
read -p "Press any key...: " ;;

# Pattern 2
2) who
read -p "Press any key...: " ;;

# Pattern 3
3) pwd
read -p "Press any key...: " ;;

# Pattern 4

4) echo "Quitting ..."

exit;;
# Default Pattern
*) echo "invalid option";;
esac
done
```

**OUTPUT**

```
Write a menu driven program to display the following options

SELECT YOUR OPTION
1. Conetnts of /etc/passwd
2. List of ouput of 'who'
3. Present working directory
4. Exit from program
Enter your menu choice [1-4]:
1
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
```

```
Write a menu driven program to display the following options

SELECT YOUR OPTION
1. Conetnts of /etc/passwd
2. List of ouput of 'who'
3. Present working directory
4. Exit from program
Enter your menu choice [1-4]:
2
macfast  tty2         2025-10-17 16:46 (tty2)
Press any key...:
```

```
Write a menu driven program to display the following options

SELECT YOUR OPTION
1. Conetnts of /etc/passwd
2. List of ouput of 'who'
3. Present working directory
4. Exit from program
Enter your menu choice [1-4]:
3
/home/macfast/mcalab
Press any key...:
```

```
Write a menu driven program to display the following options

SELECT YOUR OPTION
1. Conetnts of /etc/passwd
2. List of ouput of 'who'
3. Present working directory
4. Exit from program
Enter your menu choice [1-4]:
4
Quiting ..
macfast@macfast-V530S-07ICR:~/mcalab$
```

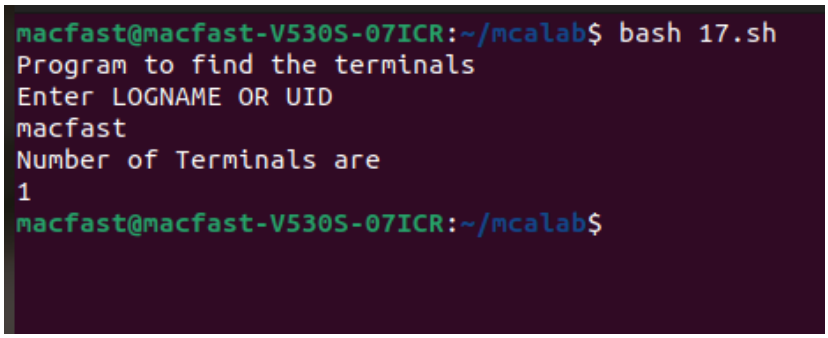
**PROGRAM 17** Date:- 23/9/2024

**AIM:-** Write a shell script to find how many terminals this user logged in.

**SOURCE CODE**

```
#!/bin/bash
echo Program to find the terminals
echo "Enter LOGNAME OR UID"
read input
# checking if input is a UID or LOGNAME
if [[ $input ]] && [ $input -eq $input 2>/dev/null ]
then
echo "Number of terminals are "
cat /etc/passwd | grep $input -c
# If input is LOGNAME
else
cat /etc/passwd>userlist
echo "Number of terminals are "
grep -c $input userlist
fi
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 17.sh
Program to find the terminals
Enter LOGNAME OR UID
macfast
Number of Terminals are
1
macfast@macfast-V530S-07ICR:~/mcalab$

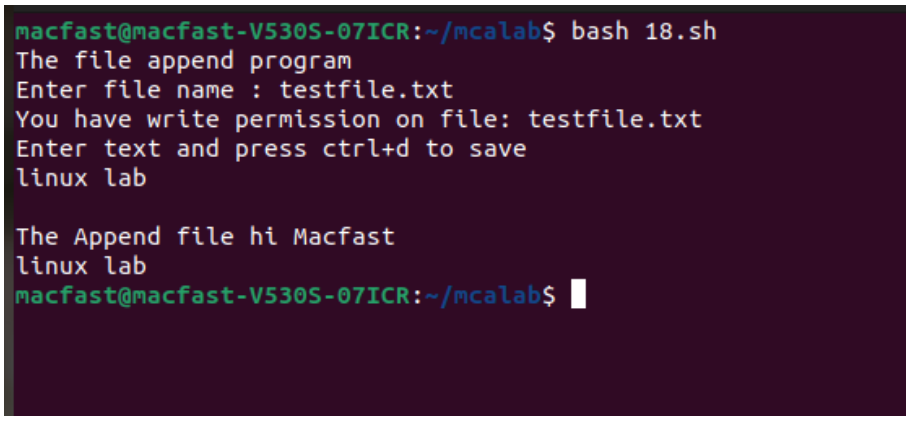
**PROGRAM 18** Date:- 26/9/2024

**AIM:-** Write a script to accept a filename while running the script and check it has the write permission, if yes prompt the user to enter a text and append the text to the given filename.

**SOURCE CODE**

```
#!/bin/bash
echo The file append program
# Taking input from user
echo -n "Enter file name : "
read file
#check the file exists or not
if test -f "$file"
then
# find out if file has write permission or not
if [ -w $file ]
then
echo you have write permission on file: $file
echo Enter text and press ctrl+d to save
cat >> $file
else
echo no write permission
fi
printf "\nThe Appended file " $file " is follows :\n"
cat $file
else
echo The file does not exists!!
fi
```

OUTPUT


macfast@macfast-V530S-07ICR:~/mcalab$ bash 18.sh
The file append program
Enter file name : testfile.txt
You have write permission on file: testfile.txt
Enter text and press ctrl+d to save
linux lab
The Append file hi Macfast
linux lab
macfast@macfast-V530S-07ICR:~/mcalab$

**PROGRAM 19** Date:- 26/9/2024

**AIM:-** Write a shell script which displays a list of all files in the current directory to which you have read, write & execute permissions.

**SOURCE CODE**

```
#!/bin/bash
echo Linux Program to List files with Permissions
for i in `ls`

do
exist=""
read=""

write=""

exe=""

#-e Returns true value if file exists.

if [ -e $i ]

then

exist=" file exists "

fi

#-r Return true value if file exists and is readable

if [ -r $i ]

then

read=" readabale "

fi

#-w Return true value if file exists and is writable

if [ -w $i ]

then

write=" Writable "

fi

#-x Return true value if file exists and is executable.

if [ -x $i ]

then

exe=" Executable "

fi

printf "%-35s" $i

echo -- $exist -- $read -- $write -- $exe

done
```

**OUTPUT**

```
mac-85@mac-85:~/Documents/gowri$ bash 19_allfile.sh
Linux Program to List files with Permissions
19_allfile.sh                       -- file exists -- readabale -- Writable --
20_samefile.sh                      -- file exists -- readabale -- Writable --
21_fileexit.sh                      -- file exists -- readabale -- Writable --
22_student_manage.sh                -- file exists -- readabale -- Writable --
27_mca.sh                           -- file exists -- readabale -- Writable --
28_file_dir_check.sh                -- file exists -- readabale -- Writable --
29.sh                               -- file exists -- readabale -- Writable --
30_file_copy.sh                     -- file exists -- readabale -- Writable --
=5000,                              -- file exists -- readabale -- Writable --
6_sell.sh                           -- file exists -- readabale -- Writable --
7_grade.sh                          -- file exists -- readabale -- Writable --
8_char_check.sh                     -- file exists -- readabale -- Writable --
bubble_23.sh                        -- file exists -- readabale -- Writable --
comb.sh                             -- file exists -- readabale -- Writable --
count_line_word.sh                  -- file exists -- readabale -- Writable --
fact_24.sh                          -- file exists -- readabale -- Writable --
fact_sum.sh                         -- file exists -- readabale -- Writable --
file1.txt                           -- file exists -- readabale -- Writable --
file3.txt                           -- file exists -- readabale -- Writable --
gross_sal.sh                        -- file exists -- readabale -- Writable --
multi.sh                            -- file exists -- readabale -- Writable --
multi.sh.txt                        -- file exists -- readabale -- Writable --
mydir                               -- file exists -- readabale -- Writable -- Executabl
myfile.txt                          -- file exists -- readabale -- Writable --
odd_pos.sh                          -- file exists -- readabale -- Writable --
power.sh                            -- file exists -- readabale -- Writable --
prime.sh                            -- file exists -- readabale -- Writable --
stud_data.dt                        -- file exists -- readabale -- Writable --
```

**PROGRAM 20** Date:- 30/9/2024

**AIM:-** Write a shell script which receives two file names as arguments.It should check whether the two file's contents are the same or not .If they are the same, delete the second file.

**SOURCE CODE**

```
#!/bin/bash

len=${#@}

if [ $len -eq 0 ]

then

echo No arguments please use with two arguments

exit

else

file1=$1

file2=$2

if [ -f $file1 -a -f $file2 ]

then

echo The file Comparison Program

#Checking the files are equal

if cmp -s "$file1" "$file2"; then

printf 'The file "%s" is the same as "%s"\n'

"$file1" "$file2"

read -p "do you want to delete the second file y/n " chk

if [ "$chk" = "y" ]

then
rm $file2
echo The second duplicate file deleted...
fi
else

printf 'The file "%s" is different from "%s"\n'

"$file1" "$file2"

fi

else

echo "File(s) not exists!"

fi

fi
```

**OUTPUT**

```
macfast@macfast-V530S-07ICR:~/mcalab$bash 20_samefile.sh myfile.txt file.txt
The file Comparison Program
The file "" is the same as ""
20_samefile.sh: line 16: myfile.txt: command not found
do you want to delete the second file y/n y
The second duplicate file deleted...
```

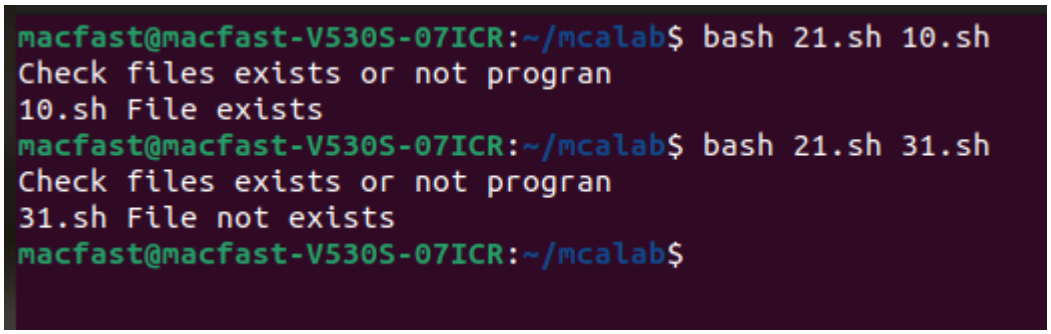
**PROGRAM 21** Date:- 30/9/2024

**AIM:-** Write a shell script, which will receive any number of filenames as arguments .The shell script should check whether such files already exist.

**SOURCE CODE**

```
#!/bin/bash

len=${#@}

if [ $len -eq 0 ]

then

echo No arguments please use with some arguments

exit

else

echo Check files exists or not Program

for i in $@

do

#-e Returns true value if file exists.

if [ -e $i ]

then

echo $i " file exists "

else

echo $i " File not Exists "

fi

done

fi
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 21.sh 10.sh
Check files exists or not progran
10.sh File exists
macfast@macfast-V530S-07ICR:~/mcalab$ bash 21.sh 31.sh
Check files exists or not progran
31.sh File not exists
macfast@macfast-V530S-07ICR:~/mcalab$

**PROGRAM 22** Date:-3/10/2024

**AIM:-** Write a shell script to perform operations for student data like view, add and delete records.

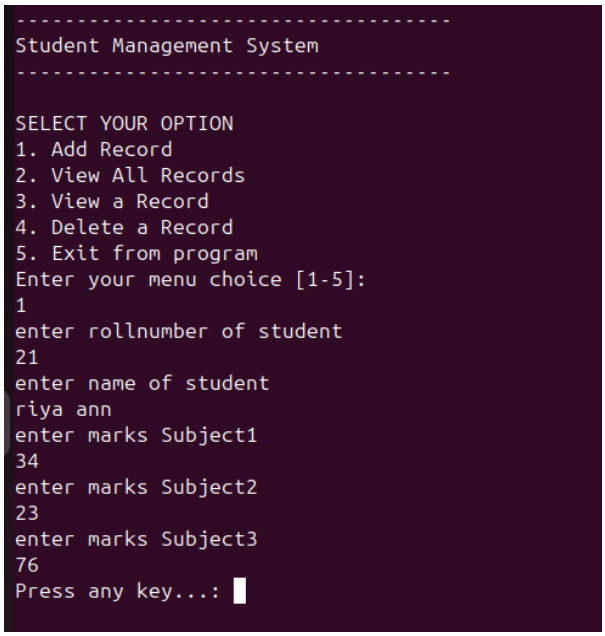
**SOURCE CODE**

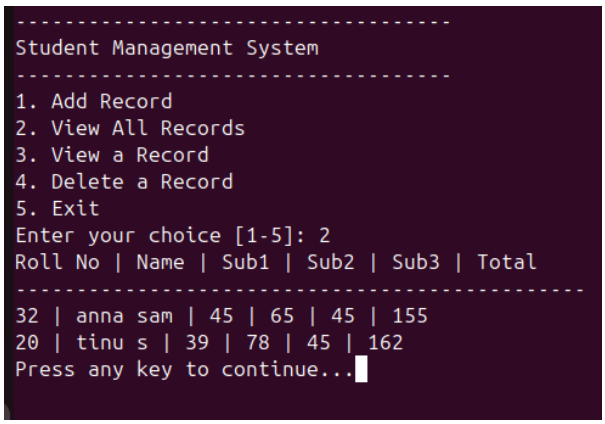
```
#!/bin/bash
while :
do
clear
echo
echo "-------------------------------------"
echo Student Management System
echo "-------------------------------------"
printf "\nSELECT YOUR OPTION\n"
echo "1. Add Record"
echo "2. View All Records"
echo "3. View a Record"
echo "4. Delete a Record"
echo "5. Exit from program"
echo "Enter your menu choice [1-5]: "
# reading choice
read choice
# case statement is used to compare one value with multiple cases.
case $choice in
# Pattern 1
1) echo "enter rollnumber of student "
read number
echo "enter name of student "
read name
echo "enter marks Subject1 "
read sub1
echo "enter marks Subject2 "
read sub2
echo "enter marks Subject3 "
```

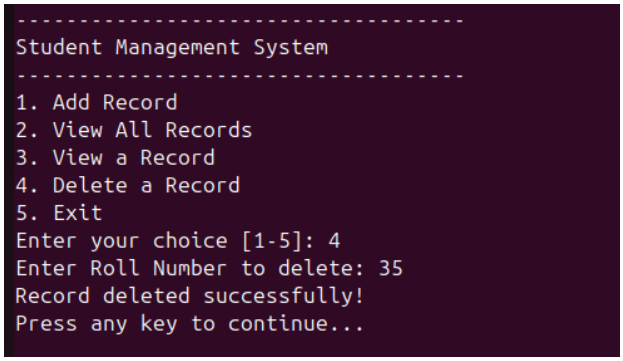
```
read sub3
total=`expr "$sub1 + $sub2 + $sub3" |bc`
record="$number,$name,$sub1,$sub2,$sub3,$total"
echo $record >> stud_data.dt
read -p "Press any key...: " ;;
# Pattern 2 view all records
2) printf "Number \tName\tSubject1\tSubject2\tSubject3\tTotal\n"
while read line
do
for j in $line
do
sno=$(echo "$j" | cut -d "," -f 1)
snm=$(echo "$j" | cut -d "," -f 2)
sub1=$(echo "$j" | cut -d "," -f 3)
sub2=$(echo "$j" | cut -d "," -f 4)
sub3=$(echo "$j" | cut -d "," -f 5)
total=$(echo "$j" | cut -d "," -f 6)
done
printf '%s\t%s\t\t%s\t%s\t\t%s\t\t%s\n' "$sno" "$snm"
"$sub1" "$sub2" "$sub3" "$total"
done < stud_data.dt
read -p "Press any key...: " ;;
# Pattern 3 view a specific record
3)
read -p "Enter the Student Number " csno
while read line
do
for j in $line
do
sno=$(echo "$j" | cut -d "," -f 1)
snm=$(echo "$j" | cut -d "," -f 2)
sub1=$(echo "$j" | cut -d "," -f 3)
```

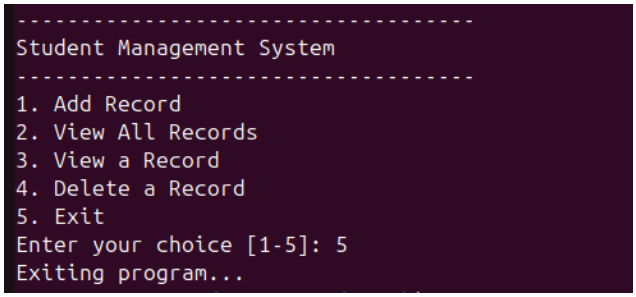
```
sub2=$(echo "$j" | cut -d "," -f 4)

sub3=$(echo "$j" | cut -d "," -f 5)

total=$(echo "$j" | cut -d "," -f 6)

done
if [ $csno -eq $sno ]
then
break
fi
done < stud_data.dt
if [ $csno -eq $sno ]
then
printf "Number \tName\tSubject1\tSubject2\tSubject3\tTotal\n"

printf '%s\t%s\t\t%s\t%s\t\t%s\t\t%s\n' "$sno" "$snm" "$sub1"

"$sub2" "$sub3" "$total"

else
echo No records found
fi
read -p "Press any key...: " ;;
# Pattern 4 Delete Record
4)
echo "enter roll number "

read rn

grep ^$rn stud_data.dt

if [ $? -ne 0 ]; then

echo "record for roll number does not exist "

else

grep -v $rn stud_data.dt > tmp

cp tmp stud_data.dt

echo "deletion complete "

fi

read -p "Press any key...: " ;;

# Pattern 5
5) echo "Quitting ..."
exit;;

# Default Pattern

*) echo "invalid option";;

esac

done
```

**OUTPUT**


Student Management System
SELECT YOUR OPTION
1. Add Record
2. View All Records
3. View a Record
4. Delete a Record
5. Exit from program
Enter your menu choice [1-5]:
1
enter rollnumber of student
21
enter name of student
riya ann
enter marks Subject1
34
enter marks Subject2
23
enter marks Subject3
76
Press any key...:


Student Management System
1. Add Record
2. View All Records
3. View a Record
4. Delete a Record
5. Exit
Enter your choice [1-5]: 2
Roll No | Name | Sub1 | Sub2 | Sub3 | Total
32 | anna sam | 45 | 65 | 45 | 155
20 | tinu s | 39 | 78 | 45 | 162
Press any key to continue...


Student Management System
1. Add Record
2. View All Records
3. View a Record
4. Delete a Record
5. Exit
Enter your choice [1-5]: 4
Enter Roll Number to delete: 35
Record deleted successfully!
Press any key to continue...


Student Management System
1. Add Record
2. View All Records
3. View a Record
4. Delete a Record
5. Exit
Enter your choice [1-5]: 5
Exiting program...

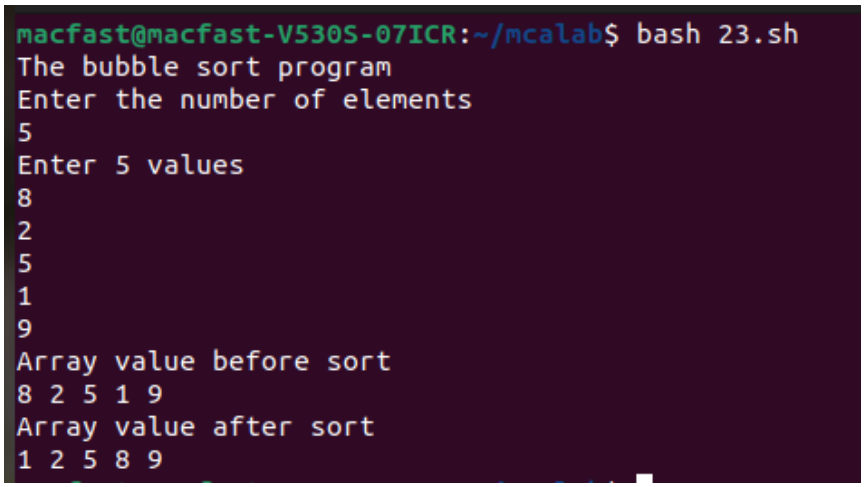
**PROGRAM 23** Date:- 3/10/2024

**AIM:-** Write a shell script to sort the given numbers in descending order using Bubble sort

**SOURCE CODE**

```
#!/bin/bash
echo The bubble sort Program
echo "enter the number of elements"
read n
echo "enter $n values"
for((i=0 ; i < $n ; i++))
do
read arr[$i]
done
echo "array values before sort"
echo ${arr[*]}
for((i=0 ; i < n ; i++))
do
for((j=0 ; j < n-i-1; j++))
do
if test ${arr[j]} -gt ${arr[((j+1))]}
then
temp=${arr[j]}
arr[$j]=${arr[$((j+1))]}
arr[$((j+1))]=$temp
fi
done
done
echo "array values after sort"
echo ${arr[*]}
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 23.sh
The bubble sort program
Enter the number of elements
5
Enter 5 values
8
2
5
1
9
Array value before sort
8 2 5 1 9
Array value after sort
1 2 5 8 9

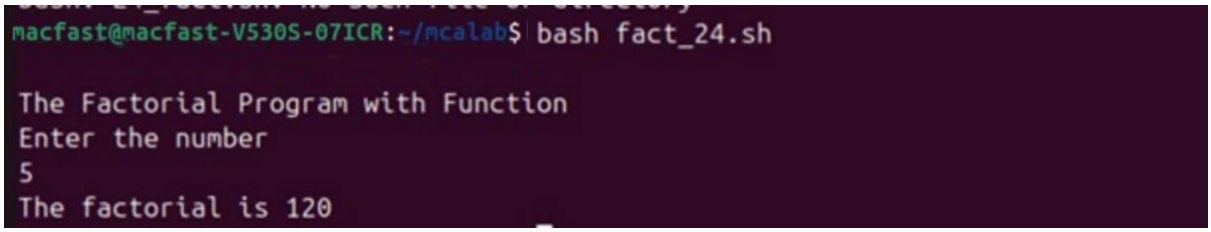
**PROGRAM 24** Date:- 7/10/2024

**AIM:-** Write a shell program to find the factorial of a number using function.

**SOURCE CODE**

```
#!/bin/bash
function fact() {
local n=$1
if [[ $n -eq 0 ]]; then
echo 1
else
echo $((n*$(fact $n-1)))
fi
}
#Factorial function
echo
echo The Factorial Program with Function
echo Enter the number
read number
echo The factorial is $(fact $number)
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash fact_24.sh
The Factorial Program with Function
Enter the number
5
The factorial is 120

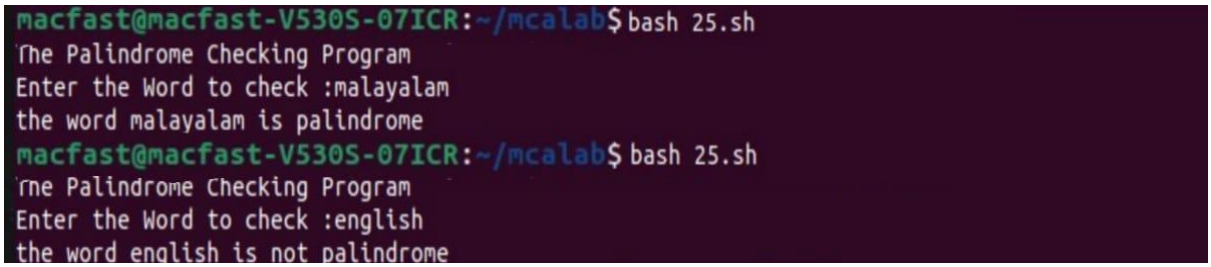
**PROGRAM 25** Date:- 7/10/2024

**AIM:-** Write a shell program to determine whether the given string is palindrome or not using function.

**SOURCE CODE**

```
#!/bin/bash
function palindrome (){
local input=$1
reverse=""
len=${#input}
for ((i=$len-1; i>=0; i-- ))
do
reverse="$reverse${input:$i:1}"
done
if [ $input == $reverse ]
then
echo the word $input is palindrome
else
echo the word $input is not palindrome
fi
}
echo The Palindrome Checking Program
read -p "Enter the Word to check " word
echo $(palindrome $word)
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 25.sh
The Palindrome Checking Program
Enter the Word to check :malayalam
the word malayalam is palindrome
macfast@macfast-V530S-07ICR:~/mcalab$ bash 25.sh
The Palindrome Checking Program
Enter the Word to check :english
the word english is not palindrome

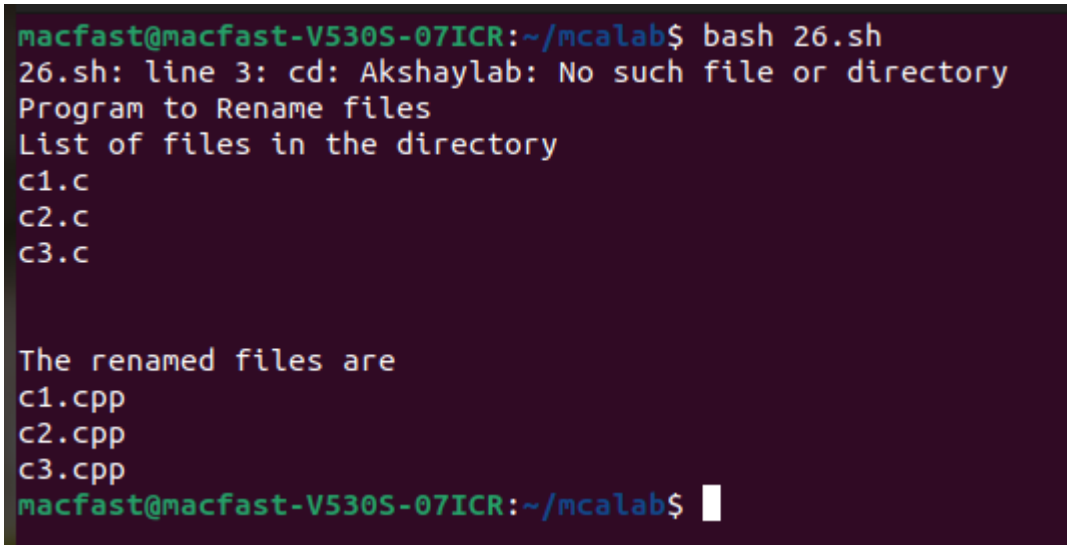
**PROGRAM 26** Date:- 14/10/2024

**AIM:-** Write a script to rename all c files to cpp files.

**SOURCE CODE**

```
#!/bin/bash
cd etc
echo Program to Rename files
echo "List of files in the directory"
for f in *.c
do
echo "$f"
done
echo
for f in *.c
do
mv "$f" "${f%.*}.cpp"
done
echo
echo The renamed files are
for f in *.cpp
do
echo "$f"
done
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 26.sh
26.sh: line 3: cd: Akshaylab: No such file or directory
Program to Rename files
List of files in the directory
c1.c
c2.c
c3.c
The renamed files are
c1.cpp
c2.cpp
c3.cpp
macfast@macfast-V530S-07ICR:~/mcalab$

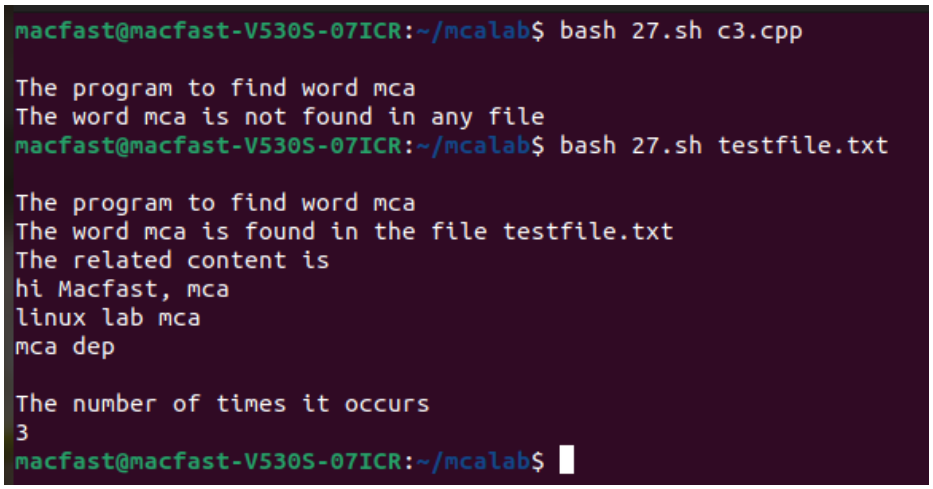
**PROGRAM 27** Date:- 14/10/2024

**AIM**:- The word “mca” is present in some of the files supplied as arguments. Write a script to search each of these files, and to stop at the first file containing the word “mca” and report it.

**SOURCE CODE**

```
#!/bin/bash
echo
echo The program to find word mca
len=${#@}
flag=0
if [ $len -eq 0 ]
then
echo No arguments please use with some arguments
exit
else
for i in $@
do
if grep -i -q "mca" $i
then
flag=1
break
fi
done
if [ $flag -eq 1 ]
then
echo "the word mca is found in the file $i"
echo The related content is
grep -i "mca" $i
echo
echo the number of times it occours
grep -c -i "mca" $i
else
echo "the word mca is not found in any file"
fi
fi
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 27.sh c3.cpp
The program to find word mca
The word mca is not found in any file
macfast@macfast-V530S-07ICR:~/mcalab$ bash 27.sh testfile.txt
The program to find word mca
The word mca is found in the file testfile.txt
The related content is
hi Macfast, mca
linux lab mca
mca dep
The number of times it occurs
3
macfast@macfast-V530S-07ICR:~/mcalab$

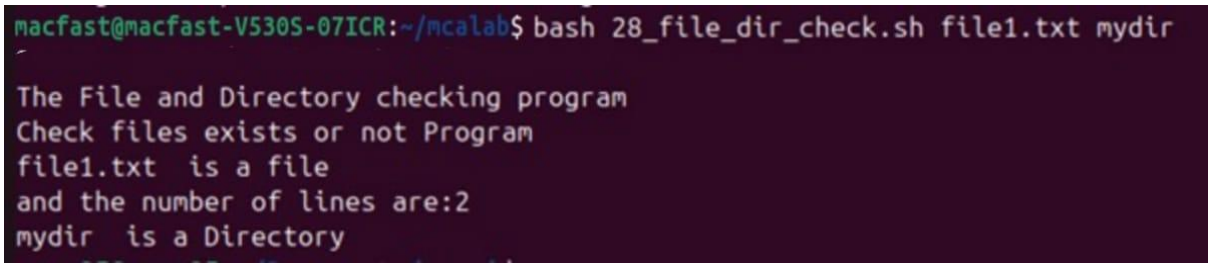
**PROGRAM 28** Date:- 17/10/2024

**AIM**:- Write a script to receive any number of filenames as arguments and to check whether the arguments supplied is a file or directory. If it is a directory, it should be appropriately reported. if it is a filename then name of the file as well as the number of lines present in it should be reported.

**SOURCE CODE**

```
#!/bin/bash
len=${#@}
echo
echo The File and Directory checking program
if [ $len -eq 0 ]
then
echo No arguments please use with some arguments
exit
else
echo Check files exists or not Program
for i in $@
do
#-e Returns true value if file exists.
if [ -d $i ]
then
echo $i " is a Directory"
else
echo $i " is a file "
echo -n and the number of lines are:
wc -l < $i
fi
done
fi
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 28_file_dir_check.sh file1.txt mydir
The File and Directory checking program
Check files exists or not Program
file1.txt  is a file
and the number of lines are:2
mydir  is a Directory

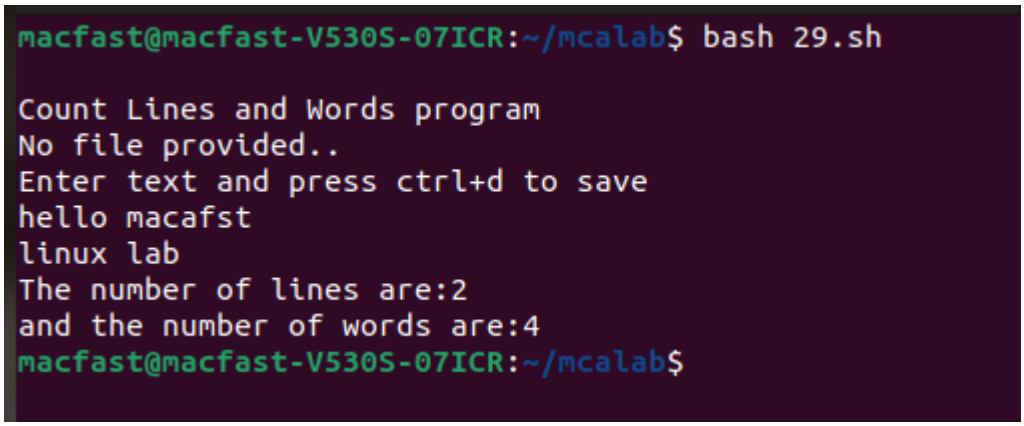
**PROGRAM 29** Date:-17/10/2024

**AIM:-** Write a script to read from a file which is supplied as a command line argument and count the number of lines and words. If there is no filename supplied, the script should accept text from the keyboard.

**SOURCE CODE**

```
#!/bin/bash

len=${#@}

echo

echo Count Lines and Words program

if [ $len -eq 0 ]

then

file="myfile.txt"

echo No files provided...

echo Enter text and press ctrl+d to save

cat >> $file

else

file=$1

fi

echo -n The number of lines are:

wc -l < $file

echo -n and the number of words are:

wc -w < $file
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 29.sh
Count Lines and Words program
No file provided..
Enter text and press ctrl+d to save
hello macafst
linux lab
The number of lines are:2
and the number of words are:4
macfast@macfast-V530S-07ICR:~/mcalab$

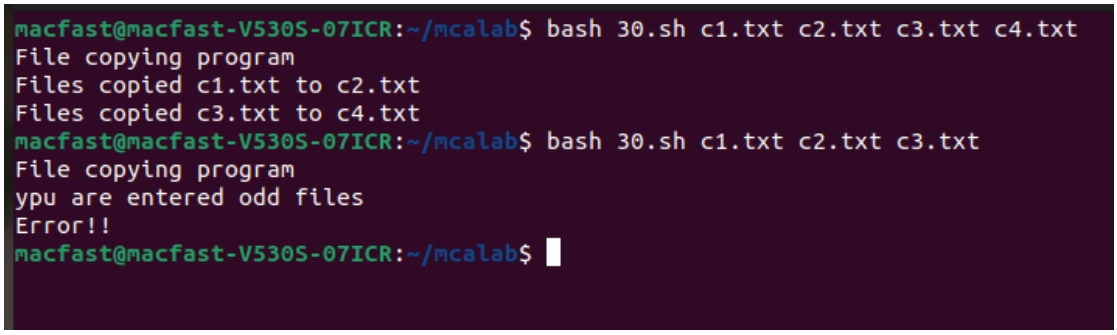
**PROGRAM 30** Date:- 21/10/2024

**AIM:-** Write a shell script which receives an even number of file names. Suppose four file names are supplied then the first file should get copied into the second file, the third file should get copied into the fourth file, and so on. If odd numbers of file names are supplied then no copying should take place and an error message should be displayed.

**SOURCE CODE**

```
#!/bin/bash
len=${#@}
echo File copying program
if [ $len -eq 0 ]
then
echo No arguments please use with some arguments
exit
elif [ `expr $len % 2` -eq 0 ]
then
for ((i=1; i<$len; i+=2))
do
if [ -f ${!i} ]
then
((j=i+1))
cp ${!i} ${!j}
echo Files Copied ${!i} to ${!j}
else
echo ${!i} " File not Exists "
fi
done
else
echo you are entered odd files
echo Error!!
fi
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 30.sh c1.txt c2.txt c3.txt c4.txt
File copying program
Files copied c1.txt to c2.txt
Files copied c3.txt to c4.txt
macfast@macfast-V530S-07ICR:~/mcalab$ bash 30.sh c1.txt c2.txt c3.txt
File copying program
ypu are entered odd files
Error!!
macfast@macfast-V530S-07ICR:~/mcalab$

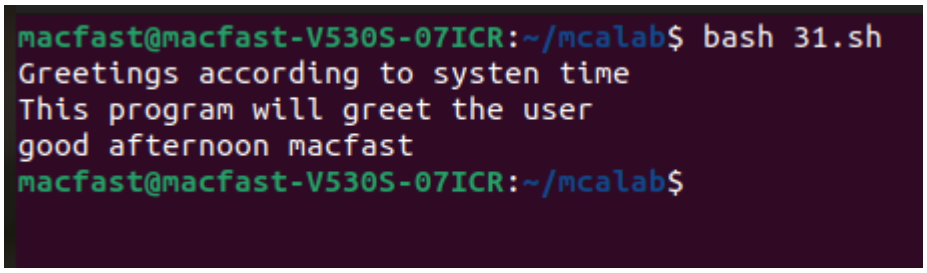
**PROGRAM 31** Date:- 21/10/2024

**AIM:-** Write a script to wish the user “Good Morning, Good Afternoon and Good Evening” when he logs in to the system based on the time.

**SOURCE CODE**

```
#!/bin/bash
echo "Greetings accordig to system time"
echo "This program will greet the user"
hour=`date +%H`
user=`whoami`
if [ $hour -lt 12 ]
then
echo good morning $user
elif [ $hour -gt 12 -a $hour -le 16 ]
then
echo good afternoon $user
elif [ $hour -gt 16 -a $hour -le 24 ]
then
echo Good night $user
fi
```

**OUTPUT**


macfast@macfast-V530S-07ICR:~/mcalab$ bash 31.sh
Greetings according to systen time
This program will greet the user
good afternoon macfast
macfast@macfast-V530S-07ICR:~/mcalab$