NETWORKING & SYSTEM ADMINISTRATION LAB

(COURSE CODE-20MCA136)

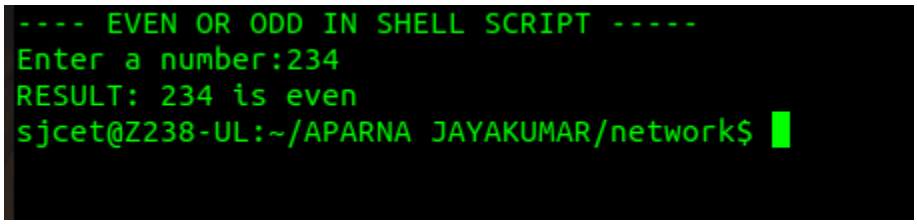
1.Practice Basic Shell Commands like:- ls, cd, du, pwd, man, cat, more, less, head, tail, mkdir,cp, mv, rm, touch, grep, sort, wc, cut, echo…Practice

- **ls -** used to display a list of content of a directory.
- **cd -** used to change the current directory.
- **du** - used to check the information of disk usage of files and directories on a system.
- **pwd -** used to display the location of the current working directory.
- **man -** used to display the user manual of any command that we can run on the terminal.
- **cat** - used as a filter, to filter a file, it is used inside pipes.
- **more** - reads files and displays the text one screen at a time.
- **less** - used for filtering and viewing text files one screen page at a time.
- **head** - prints the first 10 lines of the specified files.
- **tail** - prints the last few lines of a certain file, then terminates.
- **mkdir** - used to create a new directory under any directory.
- **cp** - used to copy a file or directory.
- **mv** - used to move a file or a directory from one location to another location.
- **rm** - used to remove a file.
- **touch** - Create a new file or update its timestamp.
- **grep** - To search for matching patterns in a file.
- **sort** - used to sort files in alphabetical order.
- **wc** - used to count the lines, words, and characters in a file.
- **cut** - used to select a specific column of a file.
- **echo** - for displaying lines of text or string which are passed as arguments on the command line

2. Write a Shell program to check if the given number is even or odd.

```
clear
echo "---- EVEN OR ODD IN SHELL SCRIPT -----"
echo -n "Enter a number:"
read n
echo -n "RESULT: "
if [ `expr $n % 2` == 0 ]
then
        echo "$n is even"
else
        echo "$n is Odd"
Fi
```

output



3. Write a Shell program to check a leap year.

```
clear
echo "LEAP YEAR SHELL SCRIPT"
echo -n "Enter a year:"
read year_checker
if [ `expr $year_checker % 4` -eq 0 ]
then
  echo "$year_checker is a leap year"
else
  echo "$year_checker is not a leap year"
fi
```

Output

```
LEAP YEAR SHELL SCRIPT
Enter a year:2014
2014 is not a leap year
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$
```

4. Write a Shell program to find the area and circumference of a circle.

```
clear
 echo "Enter the radious of the circle"
read r
area=$(echo "3.14*$r*$r" | bc  )
circum=$(echo "3.14*2*$r" | bc)
echo "area of the circle is             " $area
echo "circumference of the circle is    " $circum
```

Output

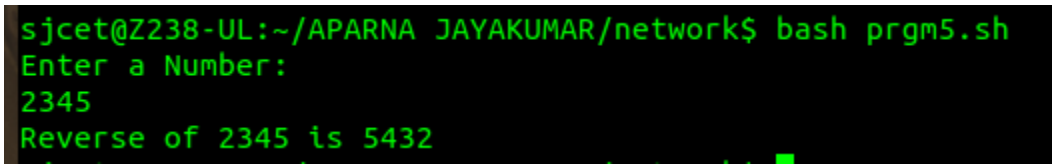
```
Enter the radious of the circle
3
area of the circle is              28.26
circumference of the circle is     18.84
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$
```

5. Write a Shell program to check the given number and its reverse are the same

```
echo "Enter a Number:"
read a
rev=0
sd=0
or=$a
while [ $a -gt 0 ]
do
sd=`expr $a % 10`
```

```
temp=`expr $rev \* 10`
rev=`expr $temp + $sd`
a=`expr $a / 10`
done
echo "Reverse of $or is $rev"
```

Output



.

6. Write a Shell program to check if the given string is palindrome or not.

```
echo "Enter a String"
read input
reverse=""

len=${#input}
for (( i=$len-1; i>=0; i-- ))
do
        reverse="$reverse${input:$i:1}"
done
if [ $input == $reverse ]
then
   echo "$input is palindrome"
else
   echo "$input is not palindrome"
fi
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm6.sh
Enter a String
aparna
aparna is not palindrome
```

7. Write a Shell program to find the sum of odd and even numbers from a set of numbers.

```
echo "enter"
read num
rev=0
even=0
odd=0
while [ $num -gt 0 ]
do
tmp=$(( $num % 10 ))
if(( $tmp % 2 == 0 ))
then
even=$(( $even + $tmp ))
else
odd=$(( $odd + $tmp ))
fi
rev=$(( $rev * 10 + $tmp ))
num=$(( $num / 10 ))
done
echo the sum of even number $even
echo the sum of odd number $odd
```

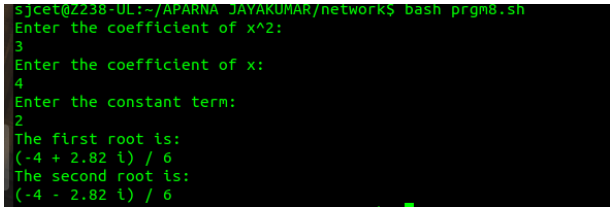
Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm7.sh
enter
123456787
the sum of even number 20
the sum of odd number 23
```

8. Write a Shell program to find the roots of a quadratic equation.

```
echo Enter the coefficient of x^2:
read a
echo Enter the coefficient of x:
read b
echo Enter the constant term:
read c
f=`echo "-($b)" |bc`
p=`expr 2 \* $a`
if [ $a -ne 0 ]
then
   d=`echo \( \( $b \* $b \) - \( 4 \* $a \* $c \) \) | bc`
   if [ $d -lt 0 ]
   then
      x=`echo "-($d)" | bc`
      s=`echo "scale=2; sqrt ( $x )" | bc`
      echo The first root is:
      echo "($f + $s i) / $p"
      echo The second root is:
      echo "($f - $s i) / $p"

   elif [ $d -eq 0 ]
   then
      res=`expr $f / $p`
      echo The root is: $res
   else
      s=`echo "scale=2; sqrt( $d )" | bc`
      res1=`echo "scale=2; ( $f + $s) / ( $p )"|bc`
      res2=`echo "scale=2; ( $f - $s) / ( $p )"|bc`
      echo The first root is: $res1
      echo The second root is: $res2
   fi
else
   echo Coefficient of x^2 can not be 0.
fi
```

Output



9. Write a Shell program to check if the given integer is an Armstrong number or not.

```
echo "Enter a number: "
read c
x=$c
sum=0
r=0
n=0
while [ $x -gt 0 ]
do
r=`expr $x % 10`
n=`expr $r \* $r \* $r \* $r`
sum=`expr $sum + $n`
x=`expr $x / 10`
done
if [ $sum -eq $c ]
then
echo "It is an Armstrong Number."
else
echo "It is not an Armstrong Number."
fi
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm9.sh
Enter a number:
1634
It is an Armstrong Number.
```

10. Write a Shell program to check if the given integer is prime or not.

```
echo -e "Enter Number : \c"
read n
while [ $n -gt 2 ]
do
for((i=2; i<=$n/2; i++))
do
  ans=$(( n%i ))
  if [ $ans -eq 0 ]
  then
    echo "$n is not a prime number."
    exit 0
  fi
done
done
echo "$n is a prime number."
```

Output

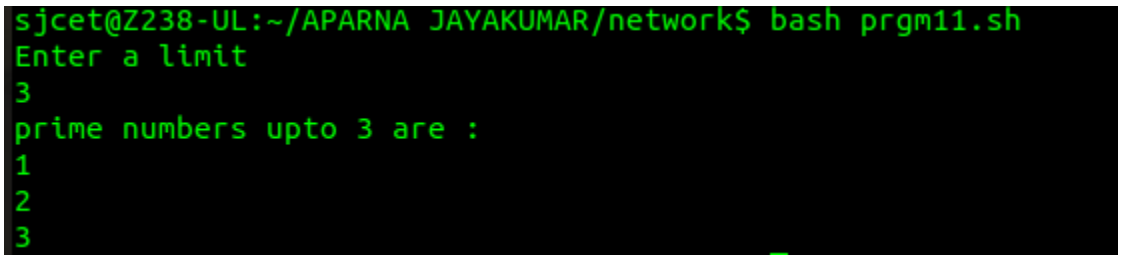
```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm10.sh
Enter Number : 12
12 is not a prime number.
```

11. Write a Shell program to generate prime numbers between 1 and 50.

```
echo "Enter a limit"
read limit
echo "prime numbers upto $limit are :"
echo "1"
```

```
i=2
while [ $i -le $limit ]
do
   flag=1
   j=2
   while [ $j -lt $i ]
   do
       rem=$(( $i % $j ))
      if [ $rem -eq 0 ]
      then
        flag=0
        break
      fi
   j=$(( $j+1 ))
   done
   if [ $flag -eq 1 ]
   then
     echo "$i"
   fi
i=$(( $i+1 ))
done
```

Output



12. Write a Shell program to find the sum of square of individual digits of a

number.echo “Enter a number:”
read n
t=$n
s=0

```
while [ $n -gt 0 ]
do
r=`expr $n % 10` s=`expr $s + $r \* $r`
n=`expr $n / 10`
done
echo “The sum of square of individual digits of $t is $s”
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm12.sh
“Enter a number:”
2324
“The sum of square of individual digits of 2324 is 33”
```

13. Write a Shell program to count the number of vowels in a line of text.

```
clear
echo "Entre a string to find the number of Vowels "
read st
len=`expr $st | wc -c`
len=`expr $len - 1`
count=0
while [ $len -gt 0 ]
do
ch=`expr $st | cut -c $len`
case $ch in

[aeiou,AEIOU]) count=`expr $count + 1` ;;
esac
len=`expr $len - 1`
done
echo "Number of vowels in the give string is $count"
```

Output

```
Entre a string to find the number of Vowels
aparna
Number of vowels in the give string is 3
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$
```

14. Write a Shell program to display student grades.

```
clear
echo ---------------------------------
echo '\tStudent Mark List'
echo ---------------------------------
echo Enter the Student name
read name
echo Enter the Register number
read rno
echo Enter the Mark1
read m1
echo Enter the Mark2
read m2
echo Enter the Mark3
read m3
echo Enter the Mark4
read m4
echo Enter the Mark5
read m5
tot=$(expr $m1 + $m2 + $m3 + $m4 + $m5)
avg=$(expr $tot / 5)
echo ---------------------------------
echo '/t Student Mark List'
echo ---------------------------------
echo "Student Name    : $name"
echo "Register Number : $rno"
echo "Mark1           : $m1"
echo "Mark2           : $m2"
echo "Mark3           : $m3"
echo "Mark4           : $m4"
echo "Mark5           : $m5"
echo "Total          : $tot"
echo "Average         : $avg"
if [ $m1 -ge 35 ] && [ $m2 -ge 35 ] && [ $m3 -ge 35 ] && [ $m4 -ge 35 ] && [ $m5
-ge 35 ]
```

```
then
 echo "Result          : Pass"

 if [ $avg -ge 90 ]
 then
  echo "Grade           : S"
 elif [ $avg -ge 80 ]
 then
        echo "Grade           : A"
 elif [ $avg -ge 70 ]
 then
        echo "Grade           : B"
 elif [ $avg -ge 60 ]
 then
        echo "Grade           : C"
 elif [ $avg -ge 50 ]
 then
        echo "Grade           : D"
 elif [ $avg -ge 35 ]
 then
        echo "Grade           : E"
 fi
else
 echo "Result          : Fail"
fi
echo ---------------------------------
```

Output

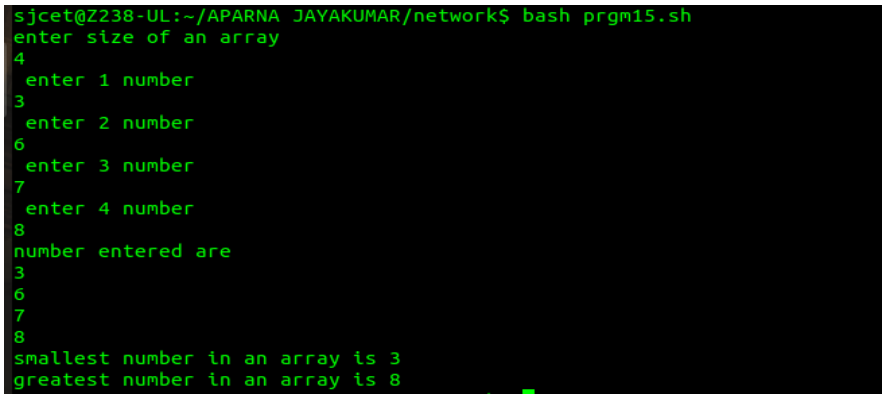
```
----------------------------------
\tStudent Mark List
----------------------------------
Enter the Student name
Aparna
Enter the Register number
1234
Enter the Mark1
38
Enter the Mark2
89
Enter the Mark3
67
Enter the Mark4
45
Enter the Mark5
89
----------------------------------
/t Student Mark List
----------------------------------
Student Name    : Aparna
Register Number : 1234
Mark1           : 38
Mark2           : 89
Mark3           : 67
Mark4           : 45
Mark5           : 89
Total           : 328
Average         : 65
Result          : Pass
Grade           : C
----------------------------------
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$
```

15. Write a Shell program to find the smallest and largest numbers from a set of numbers.

```
echo "enter size of an array"
read n
for((i=0;i<n;i++))
do
echo " enter $((i+1)) number"
read nos[$i]
done
echo "number entered are"
for((i=0;i<n;i++))
do
echo ${nos[$i]}
done
```

```
small=${nos[0]}
greatest=${nos[0]}
for((i=0;i<n;i++))
do
if [ ${nos[$i]} -lt $small ]; then
small=${nos[$i]}
elif [ ${nos[$i]} -gt $greatest ]; then
greatest=${nos[$i]}
fi
done
echo "smallest number in an array is $small"
echo "greatest number in an array is $greatest"
```

Output



16. Write a Shell program to find the smallest digit from a number.

```
echo “Enter a number:”
read n
s=9
while [ $n -gt 0 ]
do
r=`expr $n % 10`
if [ $r -lt $s ]
then
s=$r
fi
n=`expr $n / 10`
```

```
done
echo “The smallest digit is : $s”
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm16.sh
“Enter a number:”
1245
“The smallest digit is : 1”
```

17. Write a Shell program to find the sum of all numbers between 50 and 100, which are divisible by 3 and not divisible by 5.

```
for((i = 50 ; i<= 100 ; i++))
do
if [ `expr $i % 3` = 0 -a `expr $i % 5` != 0 ]
then
echo $i
fi
done
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm17.sh
51
54
57
63
66
69
72
78
81
84
87
93
96
99
```

18. Write a Shell program to find the second highest number from a set of numbers.

```
echo “Enter the number of elements:”
read n
a=0
b=0
for((i = 1 ; i <= n ; i++))
do
echo “Enter the number:”
read no
if [ $no -gt $a ]
then
b=$a a=$no
elif [ $no -gt $b ]
then
b=$no
fi
done
echo “The second highest number is : $b”
```

Output

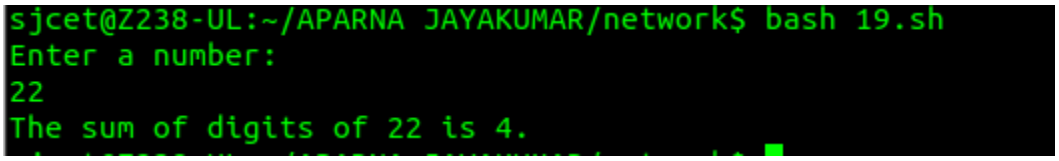
```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prrgm18.sh
“Enter the number of elements:”
4
“Enter the number:”
4
“Enter the number:”
6
“Enter the number:”
7
“Enter the number:”
8
“The second highest number is : 7”
```

19. Write a Shell program to find the sum of digits of a number using a function.

```
sum_of_digits()
{
```

```
    num=$1
    sum=0
    while [ $num -gt 0 ]
    do
       digit=$((num % 10))
       sum=$((sum + digit))
       num=$((num / 10))
    done
    echo $sum
}

echo "Enter a number: "
read num
result=$(sum_of_digits $num)
echo "The sum of digits of $num is $result."
```

Output



20. Write a Shell program to print the reverse of a number using a function.

```
echo "Enter a Number:"
read a
rev=0
sd=0
or=$a
        while [ $a -gt 0 ]OutputOutput
        do
                sd=`expr $a % 10`
                temp=`expr $rev \* 10`
                rev=`expr $temp + $sd`
                a=`expr $a / 10`
```

```
        done
echo "Reverse of $or is $rev"
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash 20.sh
Enter a Number:
34
Reverse of 34 is 43
```

21. Write a Shell program to find the factorial of a number using a for loop.

```
fact=1
echo -e "enter a number"
read n
if [ $n -le 0 ] ; then
echo "invalid number"
exit
fi
if [ $n -gt 0 ] ;
 then
for((i=$n;i>=1;i--))
do
fact=`expr $fact \* $i`
done
fi
echo "The factorial of $n is $fact"
```

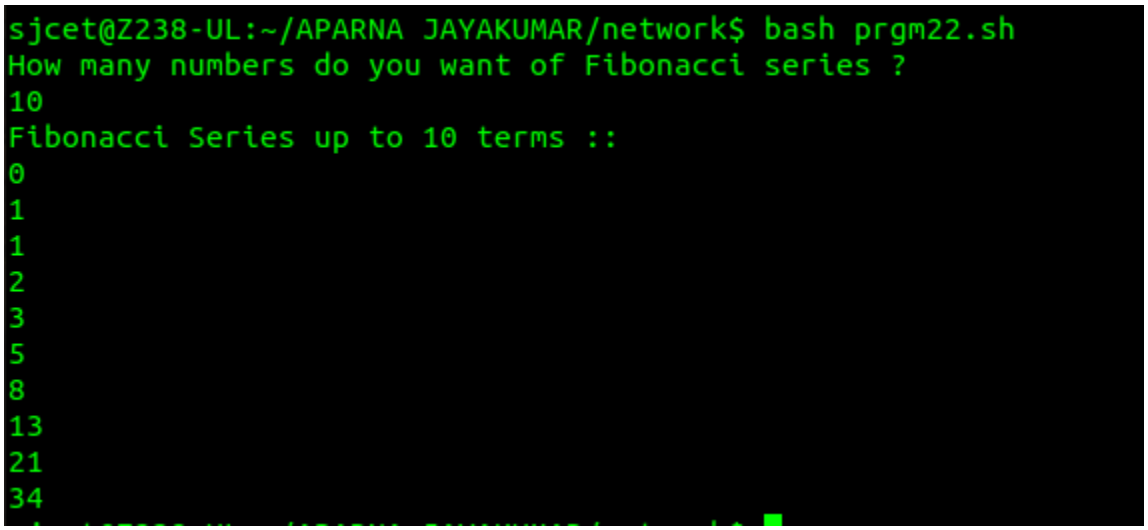
Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm21.sh
enter a number
5
The factorial of 5 is 120
```

22. Write a Shell program to generate Fibonacci series.

```
echo "How many numbers do you want from the Fibonacci series ?"
read total
x=0
y=1
i=2
echo "Fibonacci Series up to $total terms :: "
echo "$x"
echo "$y"
while [ $i -lt $total ]
do
i=`expr $i + 1 `
z=`expr $x + $y `
echo "$z"
x=$y
y=$z
done
```

Output

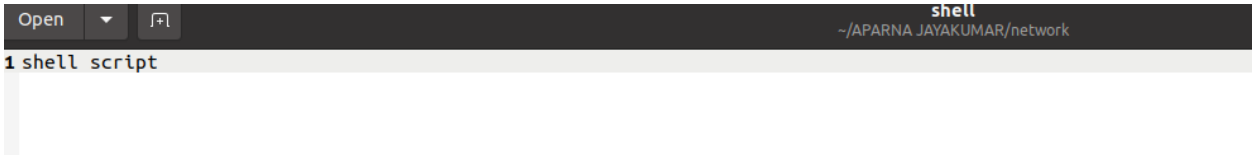


23. Write a shell script, which receives two filenames as arguments. It checks whether the two files contents are the same or not. If they are the same then the second file is deleted.

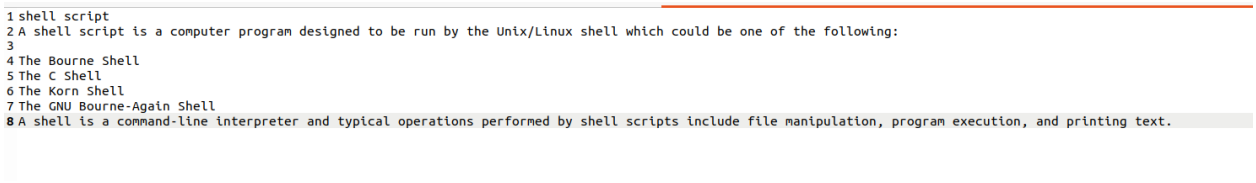
```
diff aparna shell >/dev/null
if [  `echo $?` -eq 0 ]
then
   echo Same

else
   echo "Different then remove second file"
   rm -f shell
fi
```

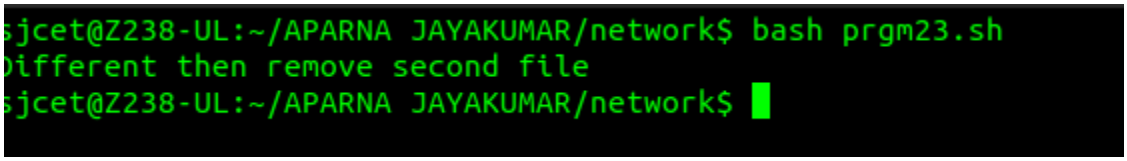
Output

Aparna



shell



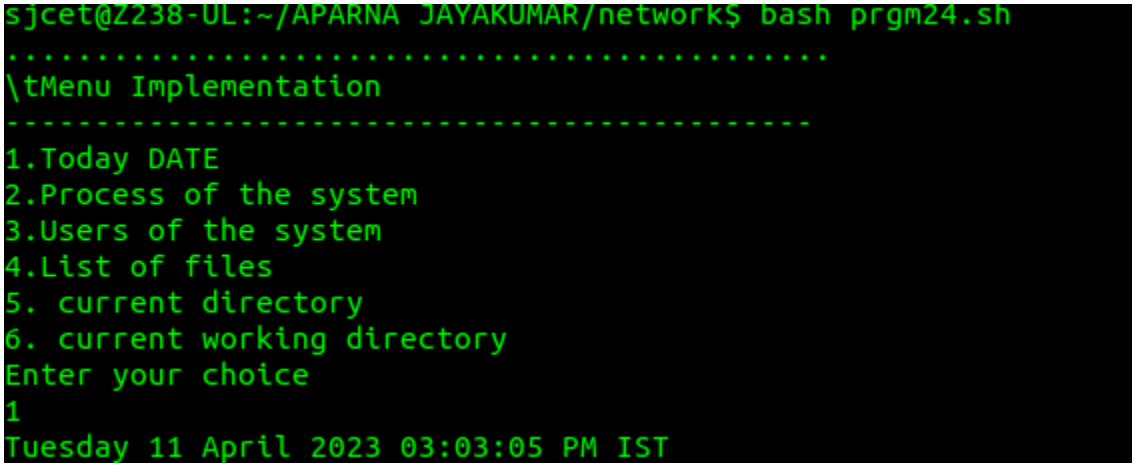


24. Write a Menu driven Shell script that Lists current directory, Prints Working Directory, displays Date and displays Users logged in

```
clear
```

```
echo .....
echo '\tMenu Implementation'
echo ----------------------------------------
echo 1.Today DATE
echo 2.Process of the system
echo 3.Users of the system
echo 4.List of files
echo 5. current directory
echo 6. current working directory
echo Enter your choice
read choice
case $choice in
        1)date;;
        2)ps;;
        3)who;;
        4)ls -1;;
        5) - [ -d "/path/dir/" ]  ;;
        6) pwd;;
        *)echo This is not a choice
esac
```

Output



25. Shell script to check executable rights for all files in the current directory, if a file does not have the execute permission then make it executable.

```
echo -n "Enter file name : "
read file
[ -w $file ] && W="Write = yes" || W="Write = No"
[ -x $file ] && X="Execute = yes" || X="Execute = No"
[ -r $file ] && R="Read = yes" || R="Read = No"
echo "$file permissions"
echo "$W"
echo "$R"
echo "$X"
```

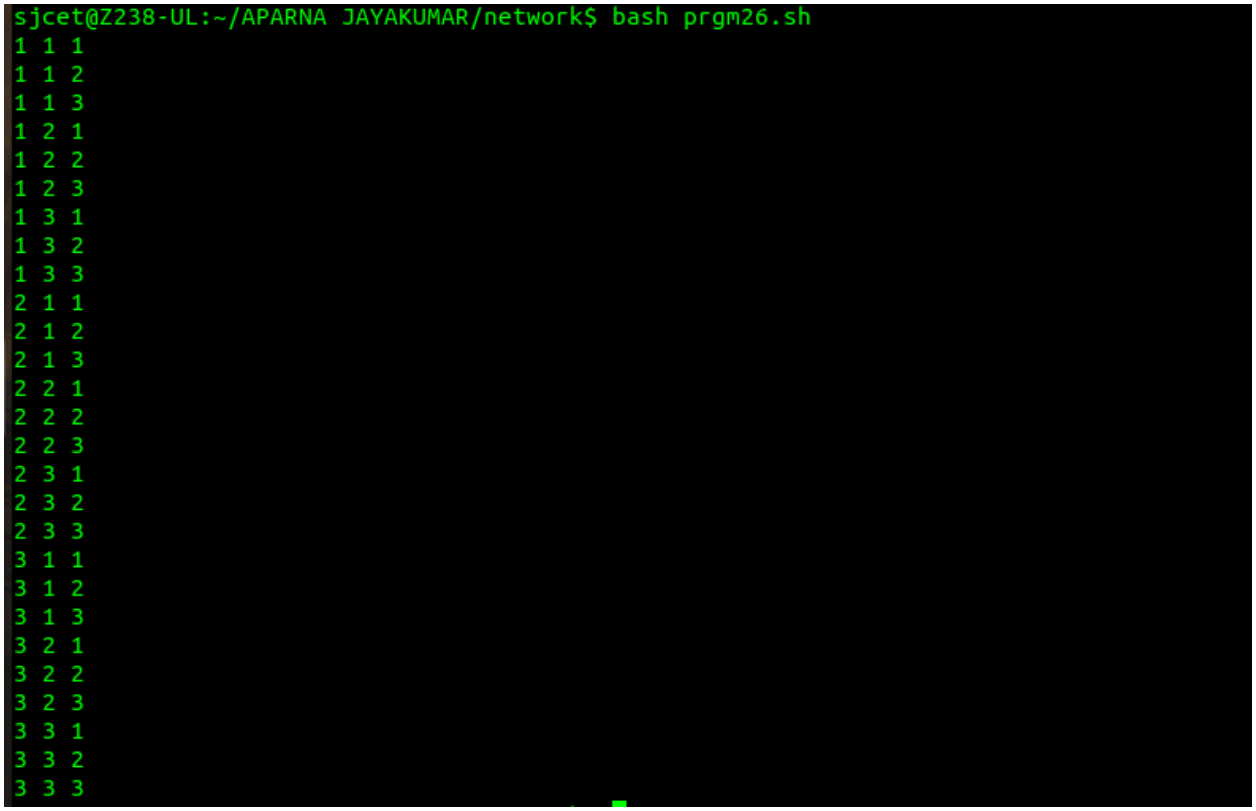
Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm25.sh
Enter file name : aparna
aparna permissions
Write = yes
Read = yes
Execute = No
```

26. Write a Shell program to generate all combinations of 1, 2, and 3 using a loop.

```
for i in 1 2 3
do
for j in 1 2 3
do
for k in 1 2 3
do
echo $i $j $k
done
done
done
```

Output



27. Write a Shell program to create the number series.

1
2 3
4 5 6
7 8 9 10

```
num=1
row=1
# Loop through rows
while [ $row -le 4 ]; do
  # Loop through numbers in row
  for (( i=1; i<=$row; i++ )); do
    # Print number and increment
    echo -n "$num "
```

```
    num=$((num+1))
  done
  # Move to next row
  echo ""
  row=$((row+1))
Done
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash 27.sh
1
2 3
4 5 6
7 8 9 10
```

28. Write a Shell program to create Pascal's triangle.

```
function binom {
  if [ $2 -eq 0 ] || [ $2 -eq $1 ]; then
    echo 1
  else
    echo $(( $(binom $(($1-1)) $(($2-1))) + $(binom $(($1-1)) $2) ))
  fi
}
echo "Enter the number of rows in Pascal's triangle: "
read rows
echo "triangle is....."
for (( i=0; i<$rows; i++ )); do
  for (( j=0; j<=$i; j++ )); do

    val=$(binom $i $j)
    echo -n "$val "
  done

  echo ""
Done
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash 28.sh
Enter the number of rows in Pascal's triangle:
4
triangle is.........
1
1 1
1 2 1
1 3 3 1
```

29. Write a Decimal to Binary Conversion Shell Script

```
echo enter number
read n
c=$(echo "obase=2;$n" | bc)
echo binary $c
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash prgm29.sh
enter number
2
binary 10
```

30. Write a Shell Script to Check Whether a String is Palindrome or not

```
echo "Enter a string: "
read string
reverse=$(echo $string | rev)
if [ "$string" == "$reverse" ]; then
  echo "$string is a palindrome."
else
  echo "$string is not a palindrome."
```

Fi

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash 30.sh
Enter a string:
aparna
aparna is not a palindrome.
```

31. Write a shell script to find out the unique words in a file and also count the occurrence of each of these words.

```
echo "Enter the file name: "
read file
if [ ! -f "$file" ]; then
  echo "File not found."
  exit 1
fi
contents=$(tr '[:upper:]' '[:lower:]' < $file | sed 's/[^a-z0-9]/ /g')
words=($contents)
declare -A count
for word in "${words[@]}"; do
  if [ -n "$word" ]; then
    ((count[$word]++))
  fi
done
echo "Unique words in $file:"
for word in "${!count[@]}"; do
  echo "$word: ${count[$word]}"
done
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash 31.sh
Enter the file name:
aparna
Unique words in aparna:
hello: 1
aparna: 1
```

32. Write a shell script to get the total count of the word “Linux” in all the “.txt” files and also across files present in subdirectories.

```
search_dir="."
# Find all ".txt" files in the search directory and its subdirectories
files=$(find "$search_dir" -type f -name "*.txt")
# Initialize the count
count=0
# Loop through each file and count the occurrences of "Linux"
for file in $files; do
  occurrences=$(grep -o "Linux" "$file" | wc -l)
  count=$((count + occurrences))
done

# Print the total count
echo "Total count of 'Linux' in all .txt files: $count"
```

Output

```
sjcet@Z238-UL:~/APARNA JAYAKUMAR/network$ bash 32.sh
Total count of 'Linux' in all .txt files: 138
```

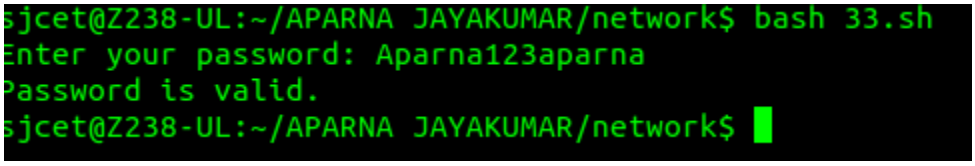
33. Write a shell script to validate password strength. Here are a few assumptions for the

password string.

Length – minimum of 8 characters.
Contain both the alphabet and number.
Include both the small and capital case letters.

```
read -p "Enter your password: " password
if [[ ${#password} -lt 8 ]]; then
   echo "Password length must be at least 8 characters."
   exit 1
fi
if ! [[ "$password" =~ [A-Za-z]+[0-9]+ ]]; then
   echo "Password must contain both alphabet and number."
   exit 1
fi
if ! [[ "$password" =~ [a-z]+ ]] || ! [[ "$password" =~ [A-Z]+ ]]; then
   echo "Passwords must include both small and capital case letters."
   exit 1
fi
echo "Password is valid."
```

OUTPUT

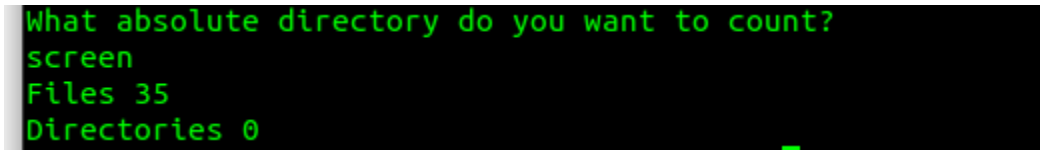


34. Write a shell script to print the count of files and subdirectories in the specified directory.

```
echo "What absolute directory do you want to count?"
read DIR
cd "$DIR" || exit
file=0
dir=0
for d in *;
do
   if [ -d "$d" ]; then
      dir=$((dir+1))
   else
```

```
        file=$((file+1))
    fi
done
echo "Files $file"
echo "Directories $dir"
```

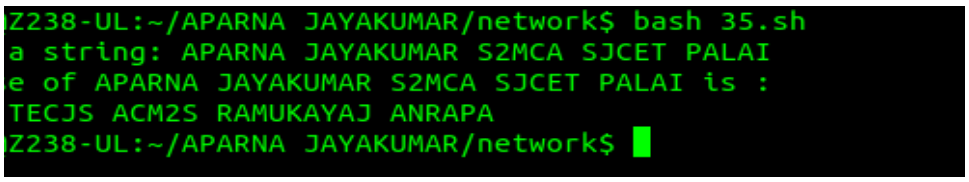
Output



35.Write a shell script to reverse the list of strings and reverse each string further in the list.

```
read -p "Enter a string: " str
length=${#str}
i=$((length-1))
while [ $i -ge 0 ]
do
    revstr=$revstr${str:$i:1}
    i=$((i-1))
done
echo "Reverse of $str is :"
echo $revstr;
```

OUTPUT



36. Installation steps for lamp

Step 1: Install Apache

1. Before installing the first LAMP component, ensure the package list on the system is up to date. In the terminal, type:

```
sudo apt update
```

2. To install the Apache package, run the following command:

```
sudo apt install apache2 -y
```

```
marko@test-main:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2
0 upgraded, 1 newly installed, 0 to remove and 11 not upgraded.
Need to get 95.6 kB of archives.
After this operation, 543 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2 amd64 2.4.41
-4ubuntu3.12 [95.6 kB]
Fetched 95.6 kB in 1s (119 kB/s)
Selecting previously unselected package apache2.
(Reading database ... 174557 files and directories currently installed.)
Preparing to unpack .../apache2_2.4.41-4ubuntu3.12_amd64.deb ...
Unpacking apache2 (2.4.41-4ubuntu3.12) ...
Setting up apache2 (2.4.41-4ubuntu3.12) ...
Processing triggers for systemd (245.4-4ubuntu3.18) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for ufw (0.36-6ubuntu1) ...
marko@test-main:~$
```

3. Check if Apache installed correctly by checking the Apache service status:

```
sudo service apache2 status
```

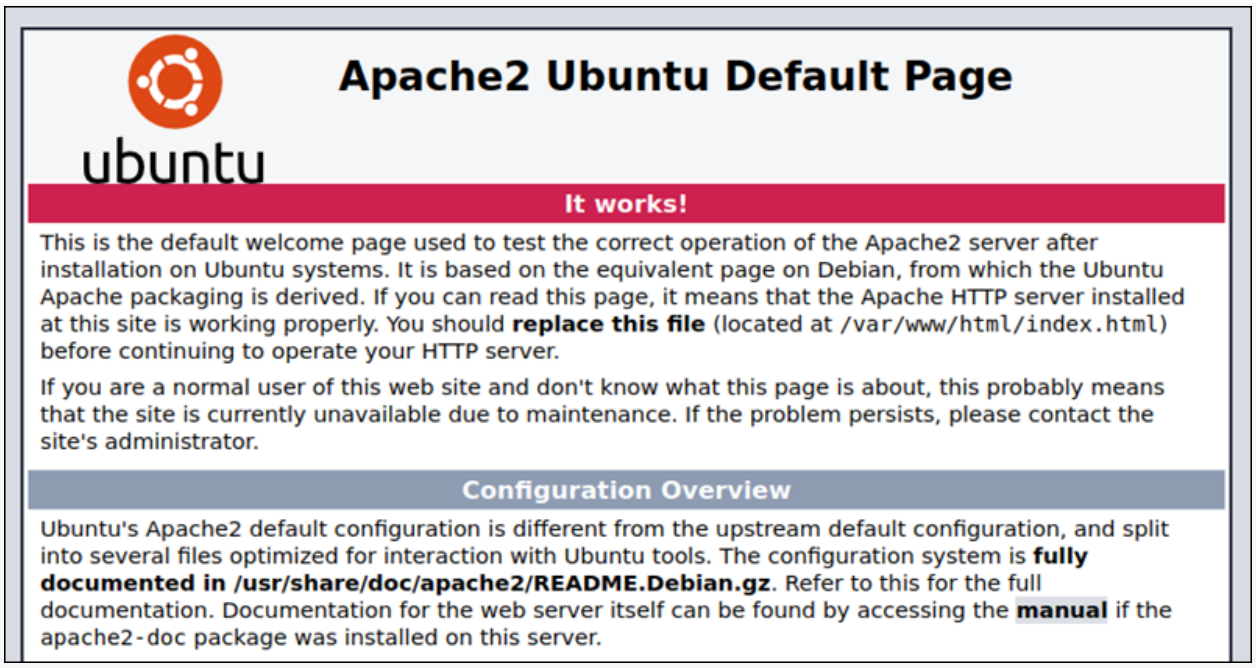
4. Next, make sure that the UFW firewall contains the Apache profiles by typing in the following command:

```
sudo ufw app list
```

5. Ensure the Apache Full profile allows the traffic on ports 80 and 443 by running the command:

```
sudo ufw app info "Apache Full"
```

6. To confirm that Apache is running, enter the IP address of your server in the address bar of an internet browser and press ENTER



Step 2: Install MySQL and Create a Database