ACN LAB

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…

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

Codes

```
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

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim eo.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash eo.sh
---- EVEN OR ODD IN SHELL SCRIPT -----
Enter a number:67
RESULT: 67 is Odd
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
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
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim leap.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash leap.sh
LEAP YEAR SHELL SCRIPT
Enter a year:2023
2023 is not a leap year
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
echo "Enter the radious of the circle"
read r
area=$(echo "3.14*$r*$r" | bc  )
circum=$(echo "3.142$r" | bc)
echo "area of the circle is             " $area
echo "circumference of the circle is    " $circum
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim ac.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash ac.sh
Enter the radious of the circle
4
area of the circle is              50.24
circumference of the circle is     3.1424
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
echo enter n
read n
num=0
while [ $n -gt 0 ]
do
num=$(expr $num \* 10)
k=$(expr $n % 10)
num=$(expr $num + $k)
n=$(expr $n / 10)
done
echo number is $num
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim reverse.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash reverse.sh
enter n
345
number is 543
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
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
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim palindrome.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash palindrome.sh
Enter a String
anina
anina is palindrome
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
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
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim sumoe.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash sumoe.sh
enter
123456789
the sum of even number 20
the sum of odd number 25
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
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
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim quadraticeq.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash quadraticeq.sh
Enter the coefficient of x^2:
2
Enter the coefficient of x:
3
Enter the constant term:
4
The first root is:
(-3 + 4.79 i) / 4
The second root is:
(-3 - 4.79 i) / 4
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
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`
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
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim amstrong.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash amstrong.sh
Enter a number:
2345
It is not an Armstrong Number.
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
echo "enter number"
read num
function prime
{
for((i=2; i<=num/2; i++))
do
  if [ $((num%i)) -eq 0 ]
  then
    echo "$num is not a prime number."
    exit
  fi
done
echo "$num is a prime number."
}
r=`prime $number`
echo "$r"
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim prime150.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash prime150.sh
enter m and n
1 50
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
echo enter m and n
read m n
for a in $(seq $m $n)
do
   k=0
   for i in $(seq 2 $(expr $a - 1))
   do
      if [ $(expr $a % $i) -eq 0 ]
      then
         k=1
         break
      fi
   done
   if [ $k -eq 0 ]
   then
   echo $a
   fi
done
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim primeornot.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash primeornot.sh
enter number
23
23 is a prime number.
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

Codes

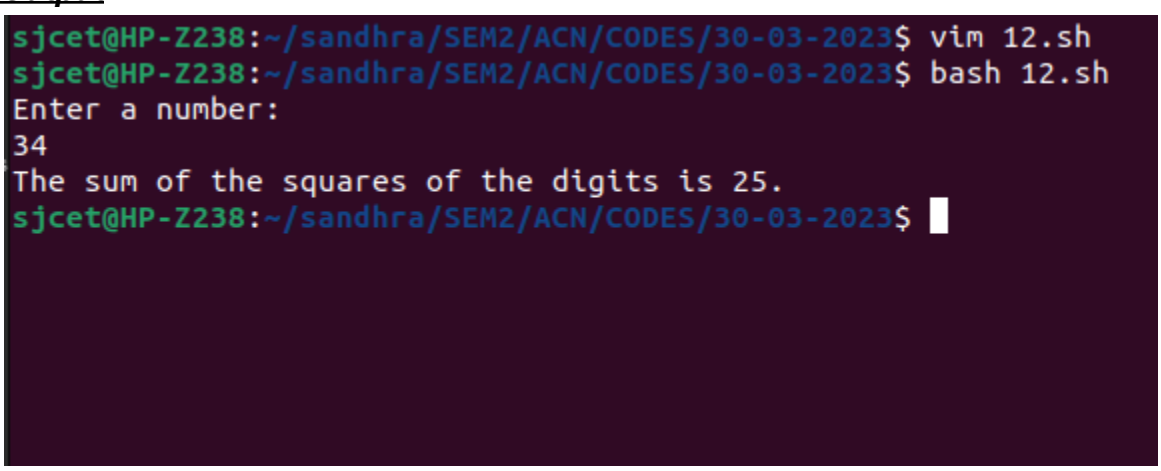
```
echo "Enter a number: "
read number

# Initialize the sum to 0
sum=0

# Loop through the digits of the number and calculate the sum of their squares
while [ $number -ne 0 ]
do
   digit=$((number % 10))
   sum=$((sum + digit * digit))
   number=$((number / 10))
done

# Output the result
echo "The sum of the squares of the digits is $sum."
```

Output



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

<u>Codes</u>

```
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"
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 13.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 13.sh
Entre a string to find the number of Vowels
celebrate
Number of vowels in the give string is 4
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

14. Write a Shell program to display student grades.

<u>Codes</u>

```
declare -A grades=(
   [Alice]=90
   [Bob]=80
   [Charlie]=70
   [David]=60
   [Emma]=50
)

# Loop through the student names and output their grades
for name in "${!grades[@]}"
do
   echo "$name: ${grades[$name]}%"
done
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 14.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 14.sh
Charlie: 70%
David: 60%
Emma: 50%
Alice: 90%
Bob: 80%
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
echo "enter the number"
read -a integers

biggest=${integers[0]}
smallest=${integers[0]}

for i in ${integers[@]}
do
    if [[ $i -gt $biggest ]]
    then
      biggest="$i"
    fi

    if [[ $i -lt $smallest ]]
    then
      smallest="$i"
    fi
done

echo "The largest number is $biggest"
echo "The smallest number is $smallest"
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 15.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 15.sh
enter the number
2 3 4 1 9 6
The largest number is 9
The smallest number is 1
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
echo "enter a: "
read a
echo "enter b : "
read b
echo "enter c : "
read c
s=$a
if [ $b -lt $s ]
then
s=$b
fi
if [ $c -lt $s ]
then
s=$c
fi
echo Smallest of $a $b $c is $s
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 16.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 16.sh
enter a:
342
enter b :
3456
enter c :
34
Smallest of 342 3456 34 is 34
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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.

<u>Codes</u>

```
sum=0
for (( num=50; num<=100; num++ ))
do

   if (( num % 3 == 0 && num % 5 != 0 )); then
      sum=$((sum + num))
   fi
done

echo "The sum of all numbers between 50 and 100, which are divisible by 3 and not divisible by 5, is $sum."
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 17.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 17.sh
The sum of all numbers between 50 and 100, which are divisible
by 3 and not divisible by 5, is 1050.
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

Codes

```
echo "Enter a set of numbers separated by spaces: "
read numbers
arr=($numbers)


sorted_arr=($(echo "${arr[@]}" | tr " " "\n" | sort -rn))


echo "The second highest number is ${sorted_arr[1]}."
```

Output

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 18.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 18.sh
Enter a set of numbers separated by spaces:
23 4 5 11 67
The second highest number is 23.
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
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."
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 19.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 19.sh
Enter a number:
234
The sum of digits of 234 is 9.
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
reverse_number() {
   num=$1
   rev=0
   while [ $num -gt 0 ]
   do
      digit=$((num % 10))
      rev=$((rev * 10 + digit))
      num=$((num / 10))
   done
   echo $rev
}
echo "Enter a number: "
read num

result=$(reverse_number $num)


echo "The reverse of $num is $result."
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 20.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 20.sh
Enter a number:
34567
The reverse of 34567 is 76543.
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
echo "Enter a number: "
read num
factorial=1

for (( i=1; i<=$num; i++ ))
do
   factorial=$((factorial * i))
done


echo "The factorial of $num is $factorial."
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 21.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 21.sh
Enter a number:
8
The factorial of 8 is 40320.
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

22. Write a Shell program to generate Fibonacci series.

<u>Codes</u>

```
echo "Enter the number of terms to generate: "
read num

# Initialize the first two terms of the series
a=0
b=1

# Output the first two terms
echo -n "$a $b"

# Generate the rest of the series using a loop
for (( i=3; i<=$num; i++ ))
do
   # Calculate the next term
   c=$((a + b))

   # Output the next term
   echo -n " $c"

   # Shift the values of a and b to prepare for the next iteration
   a=$b
   b=$c
done

echo
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 22.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 22.sh
Enter the number of terms to generate:
10
0 1 1 2 3 5 8 13 21 34
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
if [ $# -ne 2 ]
then
  echo "Usage: $0 file1 file2"
  exit 1
fi

if [ ! -f "$1" ] || [ ! -f "$2" ]
then
  echo "Error: File does not exist."
  exit 1
fi

if cmp -s "$1" "$2"
then
  echo "Files have identical contents. Deleting $2"
  rm "$2"
else
  echo "Files have different contents."
fi
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 23.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 23.sh
Usage: 23.sh file1 file2
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
echo "Select an option:"
echo "1. List current directory"
echo "2. Print working directory"
echo "3. Display date"
echo "4. Display users logged in"

read option

case $option in
  1)
    ls -l
    ;;
  2)
    pwd
    ;;
  3)
    date
    ;;
  4)
    who
    ;;
  *)
    echo "Invalid option selected"
    ;;
esac
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 24.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 24.sh
Select an option:
1. List current directory
2. Print working directory
3. Display date
4. Display users logged in
2
/home/sjcet/sandhra/SEM2/ACN/CODES/30-03-2023
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 24.sh
Select an option:
1. List current directory
2. Print working directory
3. Display date
4. Display users logged in
3
Tuesday 11 April 2023 02:54:40 PM IST
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 24.sh
Select an option:
1. List current directory
2. Print working directory
3. Display date
4. Display users logged in
4
sjcet    tty2         2023-04-11 14:21 (tty2)
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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.

<u>Codes</u>

```
find . -type f | while read file; do

  if [ ! -x "$file" ]; then

    chmod +x "$file"
    echo "Made $file executable"
  fi
done
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 25.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 25.sh
Made ./19.sh executable
Made ./14.sh executable
Made ./7.sh executable
Made ./11.sh executable
Made ./20.sh executable
Made ./8.sh executable
Made ./3.sh executable
Made ./22.sh executable
Made ./33.sh executable
Made ./13.sh executable
Made ./5.sh executable
Made ./27.sh executable
Made ./2.sh executable
Made ./23.sh executable
Made ./28.sh executable
Made ./4.sh executable
Made ./21.sh executable
Made ./34.sh executable
Made ./29.sh executable
Made ./31.sh executable
Made ./32.sh executable
Made ./15.sh executable
Made ./10.sh executable
Made ./26.sh executable
Made ./30.sh executable
Made ./18.sh executable
Made ./6.sh executable
Made ./12.sh executable
Made ./17.sh executable
Made ./9.sh executable
Made ./24.sh executable
Made ./25.sh executable
Made ./16.sh executable
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
arr=(1 2 3)
for i in "${arr[@]}"; do
        for j in "${arr[@]}"; do
                for k in "${arr[@]}"; do
                        echo "$i$j$k"
                done
        done
done
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 26.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 26.sh
111
112
113
121
122
123
131
132
133
211
212
213
221
222
223
231
232
233
311
312
313
321
322
323
331
332
333
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

27. Write a Shell program to create the number series.
1
2 3
4 5 6
7 8 9 10

<u>Codes</u>

```
count=1
rows=4

for ((i=1; i<=rows; i++))
do
  for ((j=1; j<=i; j++))
  do
    echo -n "$count "
    count=$((count+1))
  done
  echo ""
done
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 27.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 27.sh
1
2 3
4 5 6
7 8 9 10
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
function binom {
  if [ $2 -eq 0 ] || [ $2 -eq $1 ]; then
    echo 1
  else
    echo $(( $(binom $(($1-1)) $(($2-1))) + $(binom $(($1-1)) $2) ))
  fi
}

# Get the number of rows from the user
echo "Enter the number of rows in Pascal's triangle: "
read rows

# Loop through each row
for (( i=0; i<$rows; i++ )); do
  # Loop through each element in the row
  for (( j=0; j<=$i; j++ )); do
    # Calculate the binomial coefficient and print
    val=$(binom $i $j)
    echo -n "$val "
  done
  # Move to next row
  echo ""
done
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 28.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 28.sh
Enter the number of rows in Pascal's triangle:
6
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

29. Write a Decimal to Binary Conversion Shell Script

Codes

```
echo "Enter a decimal number: "
read decimal
binary=""
while [ $decimal -gt 0 ]; do
  remainder=$((decimal % 2))
  binary="$remainder$binary"
  decimal=$((decimal / 2))
done
echo "The binary equivalent is: $binary"
```

Output

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 29.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 29.sh
Enter a decimal number:
77
The binary equivalent is: 1001101
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
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
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 30.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 30.sh
Enter a string:
malayalam
malayalam is a palindrome.
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
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
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 31.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 31.sh
Enter the file name:
29.sh
Unique words in 29.sh:
the: 5
enter: 1
decimal: 8
is: 1
done: 1
do: 1
while: 1
echo: 2
2: 2
0: 1
a: 1
bash: 1
for: 1
number: 4
prompt: 1
bin: 1
convert: 2
to: 2
equivalent: 1
binary: 7
gt: 1
remainder: 2
print: 1
user: 1
read: 1
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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.

Codes

```
search_dir="."
files=$(find "$search_dir" -type f -name "*.txt")
count=0
for file in $files; do
  occurrences=$(grep -o "Linux" "$file" | wc -l)
  count=$((count + occurrences))
done
echo "Total count of 'Linux' in all .txt files: $count"
```

Output

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 32.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 32.sh
Total count of 'Linux' in all .txt files: 0
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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 alphabet and number. Include both the small and capital case letters.

<u>Codes</u>

```
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 "Password must include both small and capital case letters."
   exit 1
fi

echo "Password is valid.
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 33.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 33.sh
Enter your password: ss345@1345
Password must include both small and capital case letters.
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

Codes

```
if [ $# -eq 0 ]; then
    echo "Usage: $0 directory"
    exit 1
fi

directory=$1

if [ ! -d $directory ]; then
    echo "Error: $directory is not a directory"
    exit 1
fi

num_files=$(find $directory -maxdepth 1 -type f | wc -l)
num_dirs=$(find $directory -maxdepth 1 -type d | wc -l)

echo "Number of files in $directory: $num_files"
echo "Number of directories in $directory: $((num_dirs - 1))"
```

Output

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 34.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 34.sh
Usage: 34.sh directory
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```

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

<u>Codes</u>

```
my_list=("string1" "string2" "string3" "string4")


my_list=($(echo "${my_list[@]}" | tr ' ' '\n' | tac | tr '\n' ' '))

for i in "${!my_list[@]}"
do
  my_list[$i]=`echo ${my_list[$i]} | rev`
done

echo "${my_list[@]}"
```

<u>Output</u>

```
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ vim 35.sh
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$ bash 35.sh
4gnirts 3gnirts 2gnirts 1gnirts
sjcet@HP-Z238:~/sandhra/SEM2/ACN/CODES/30-03-2023$
```