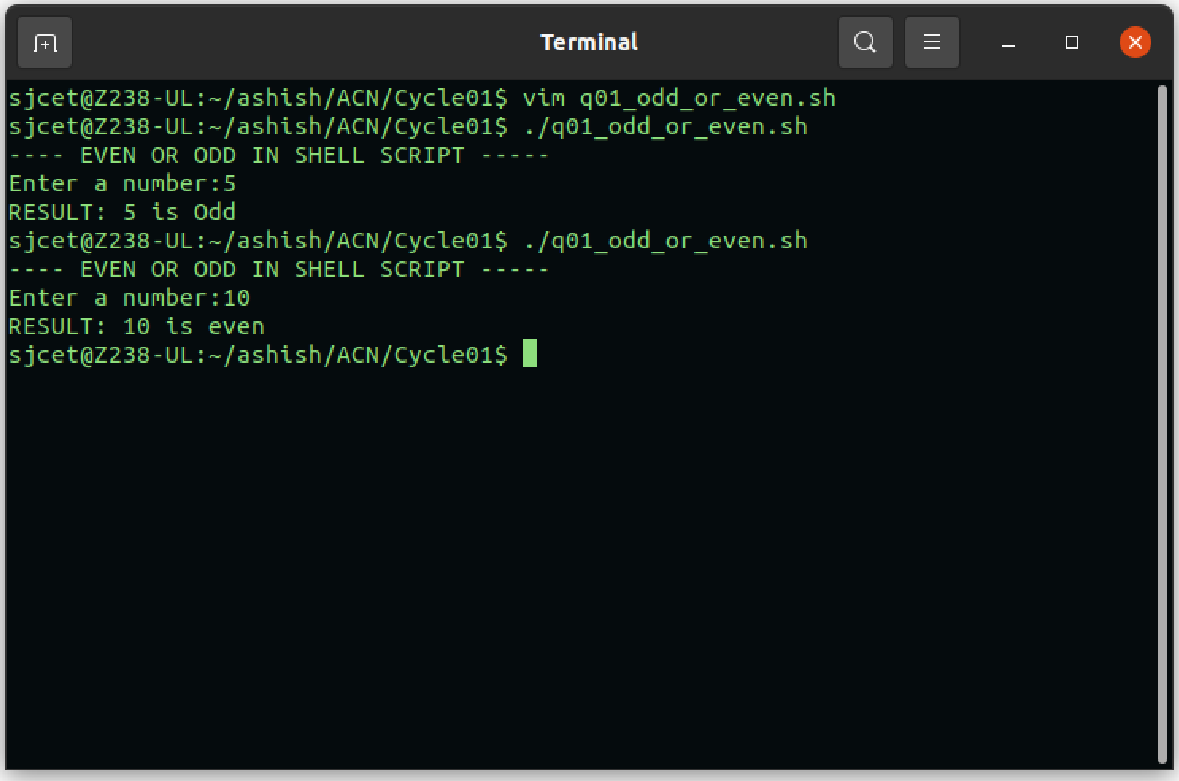
NETWORKING LAB

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.

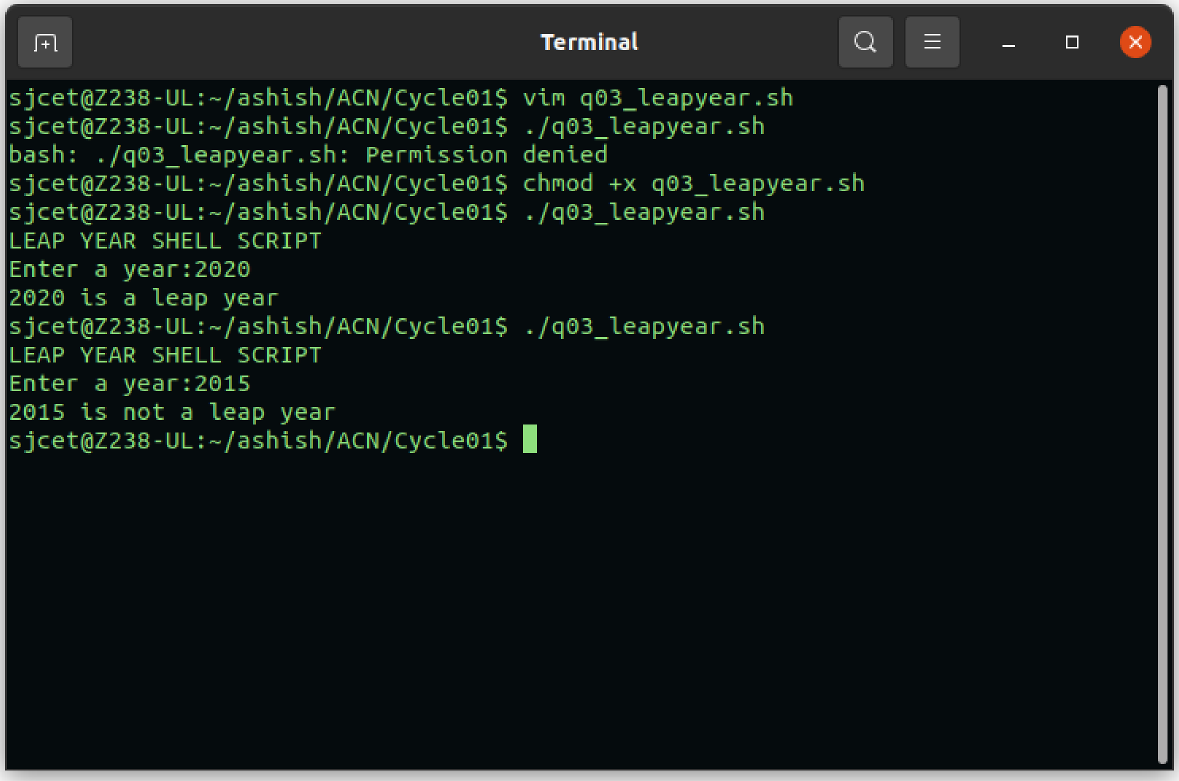
```
echo "---- EVEN OR ODD IN SHELL SCRIPT -----"
echo -n "Enter a number:"
read n
echo -n "RESULT: "
if [ `expr $n % 2` == 0 ]
then
        echo "$n is even"
else
        echo "$n is Odd"
fi
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q01_odd_or_even.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q01_odd_or_even.sh
---- EVEN OR ODD IN SHELL SCRIPT -----
Enter a number:5
RESULT: 5 is Odd
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q01_odd_or_even.sh
---- EVEN OR ODD IN SHELL SCRIPT -----
Enter a number:10
RESULT: 10 is even
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

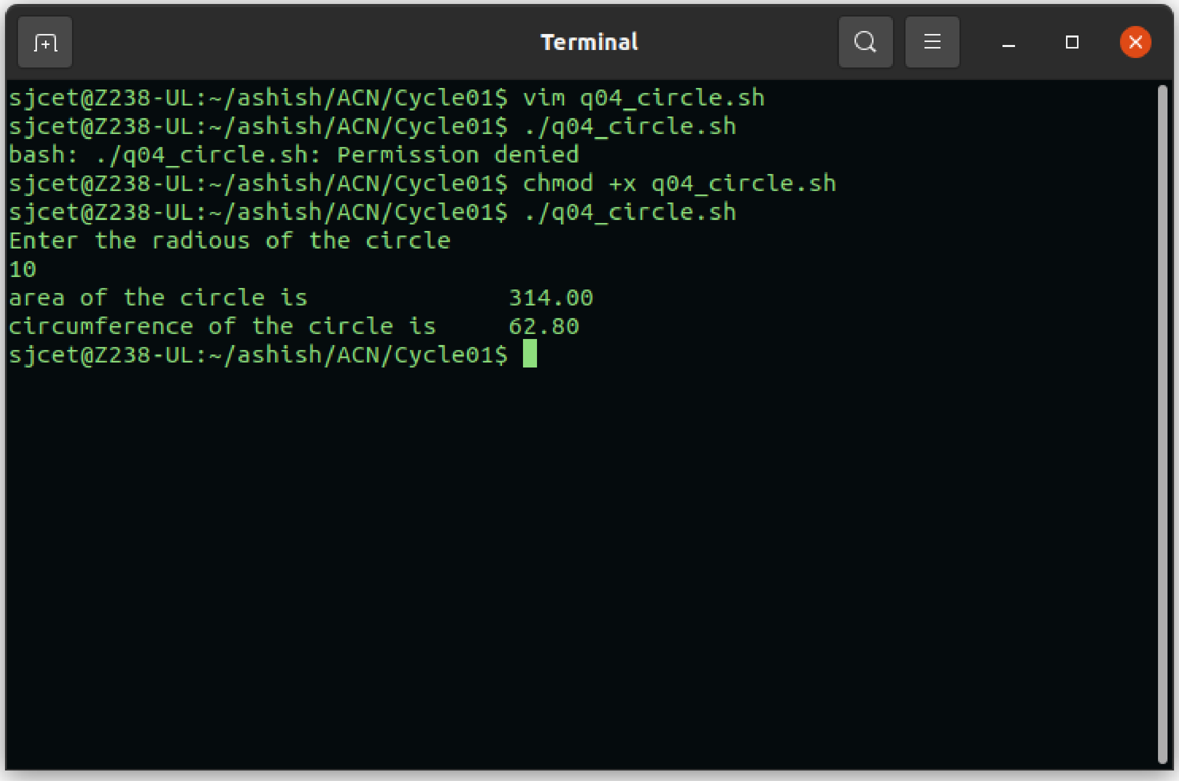
```
echo "LEAP YEAR SHELL SCRIPT"
echo -n "Enter a year:"
read year_checker
if [ `expr $year_checker % 4` -eq 0 ]
then
        echo "$year_checker is a leap year"
else
        echo "$year_checker is not a leap year"
fi
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q03_leapyear.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q03_leapyear.sh
bash: ./q03_leapyear.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q03_leapyear.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q03_leapyear.sh
LEAP YEAR SHELL SCRIPT
Enter a year:2020
2020 is a leap year
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q03_leapyear.sh
LEAP YEAR SHELL SCRIPT
Enter a year:2015
2015 is not a leap year
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

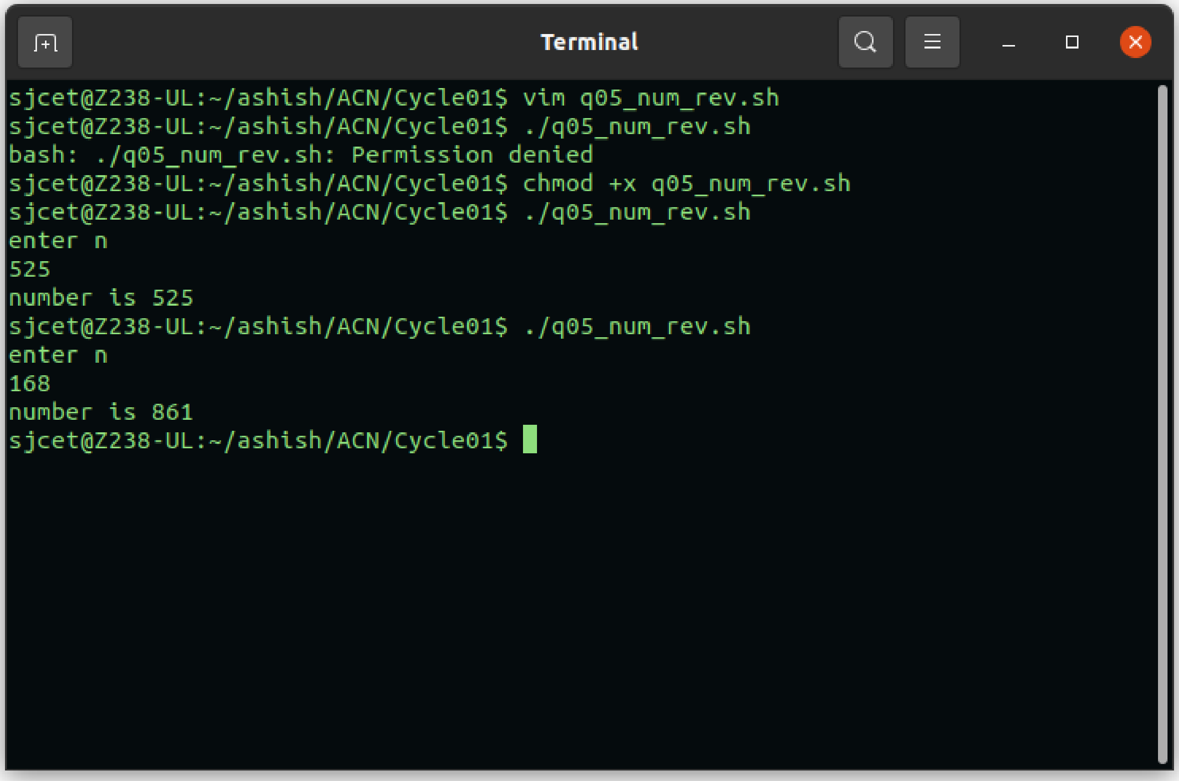
```
  echo "Enter the radius of the circle"
read r
area=$(echo "3.14*$r*$r" | bc  )
circum=$(echo "3.14*2*$r" | bc)
echo "area of the circle is            " $area
echo "circumference of the circle is    " $circum
~
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q04_circle.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q04_circle.sh
bash: ./q04_circle.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q04_circle.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q04_circle.sh
Enter the radious of the circle
10
area of the circle is              314.00
circumference of the circle is     62.80
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

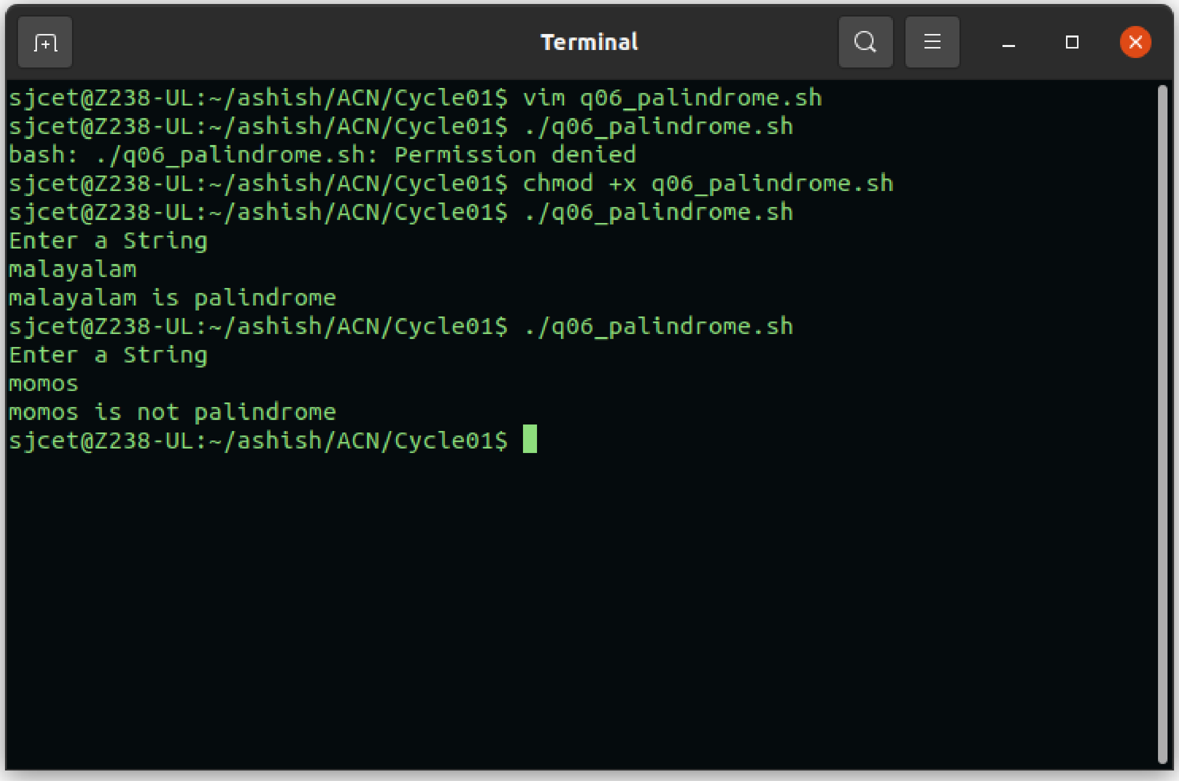
```
echo enter n
read n
num=0
while [ $n -gt 0 ]
do
num=$(expr $num \* 10)
k=$(expr $n % 10)
num=$(expr $num + $k)
n=$(expr $n / 10)
done
echo number is $num
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q05_num_rev.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q05_num_rev.sh
bash: ./q05_num_rev.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q05_num_rev.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q05_num_rev.sh
enter n
525
number is 525
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q05_num_rev.sh
enter n
168
number is 861
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

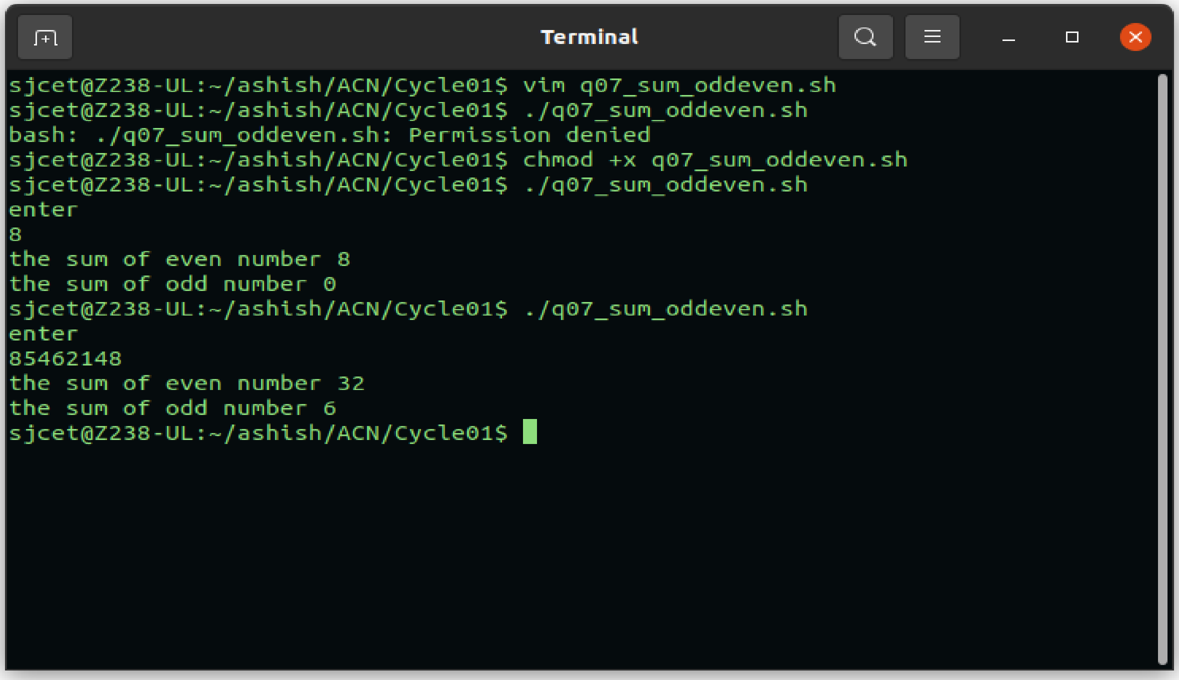
```
echo "Enter a String"
read input
reverse=""

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
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q06_palindrome.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q06_palindrome.sh
bash: ./q06_palindrome.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q06_palindrome.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q06_palindrome.sh
Enter a String
malayalam
malayalam is palindrome
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q06_palindrome.sh
Enter a String
momos
momos is not palindrome
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

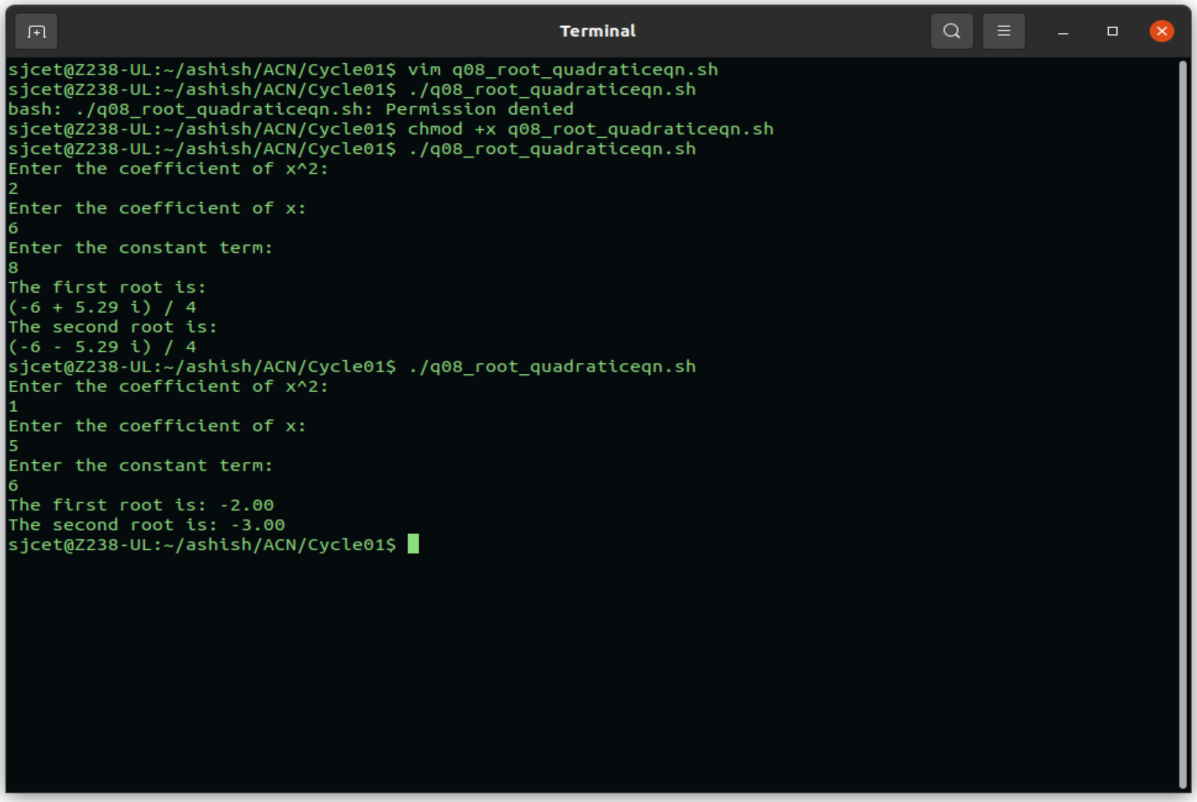
```
echo "enter"
read num
rev=0
even=0
odd=0
while [ $num -gt 0 ]
do
tmp=$(( $num % 10 ))
if(( $tmp % 2 == 0 ))
then
even=$(( $even + $tmp ))
else
odd=$(( $odd + $tmp ))
fi
rev=$(( $rev * 10 + $tmp ))
num=$(( $num / 10 ))
done
echo the sum of even number $even
echo the sum of odd number $odd
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q07_sum_oddeven.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q07_sum_oddeven.sh
bash: ./q07_sum_oddeven.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q07_sum_oddeven.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q07_sum_oddeven.sh
enter
8
the sum of even number 8
the sum of odd number 0
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q07_sum_oddeven.sh
enter
85462148
the sum of even number 32
the sum of odd number 6
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

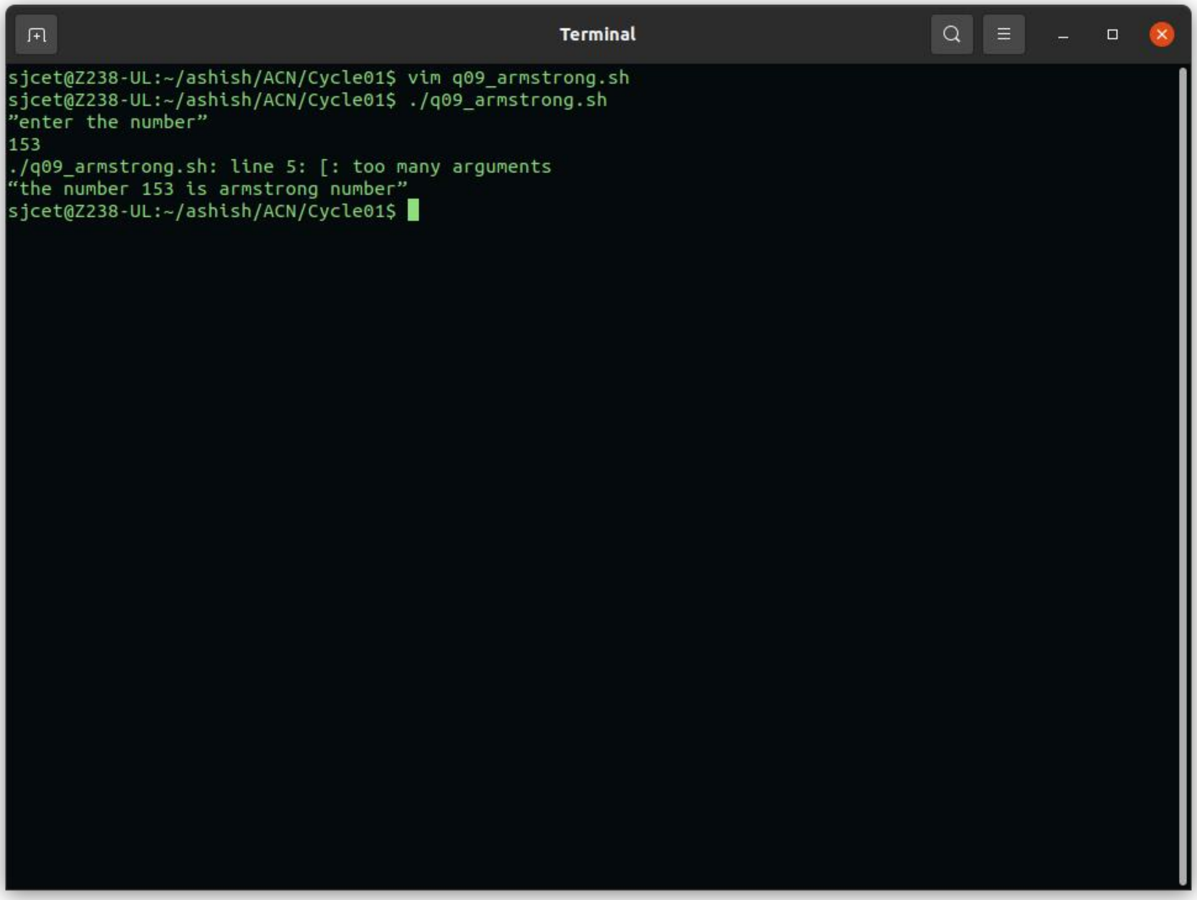
```
echo Enter the coefficient of x^2:
read a
echo Enter the coefficient of x:
read b
echo Enter the constant term:
read c
f=`echo "-($b)" |bc`
p=`expr 2 \* $a`
if [ $a -ne 0 ]
then
    d=`echo \( \( $b \* $b \) - \( 4 \* $a \* $c \) \) | bc`
    if [ $d -lt 0 ]
    then
        x=`echo "-($d)" | bc`
        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
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q08_root_quadraticeqn.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q08_root_quadraticeqn.sh
bash: ./q08_root_quadraticeqn.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q08_root_quadraticeqn.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q08_root_quadraticeqn.sh
Enter the coefficient of x^2:
2
Enter the coefficient of x:
6
Enter the constant term:
8
The first root is:
(-6 + 5.29 i) / 4
The second root is:
(-6 - 5.29 i) / 4
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q08_root_quadraticeqn.sh
Enter the coefficient of x^2:
1
Enter the coefficient of x:
5
Enter the constant term:
6
The first root is: -2.00
The second root is: -3.00
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

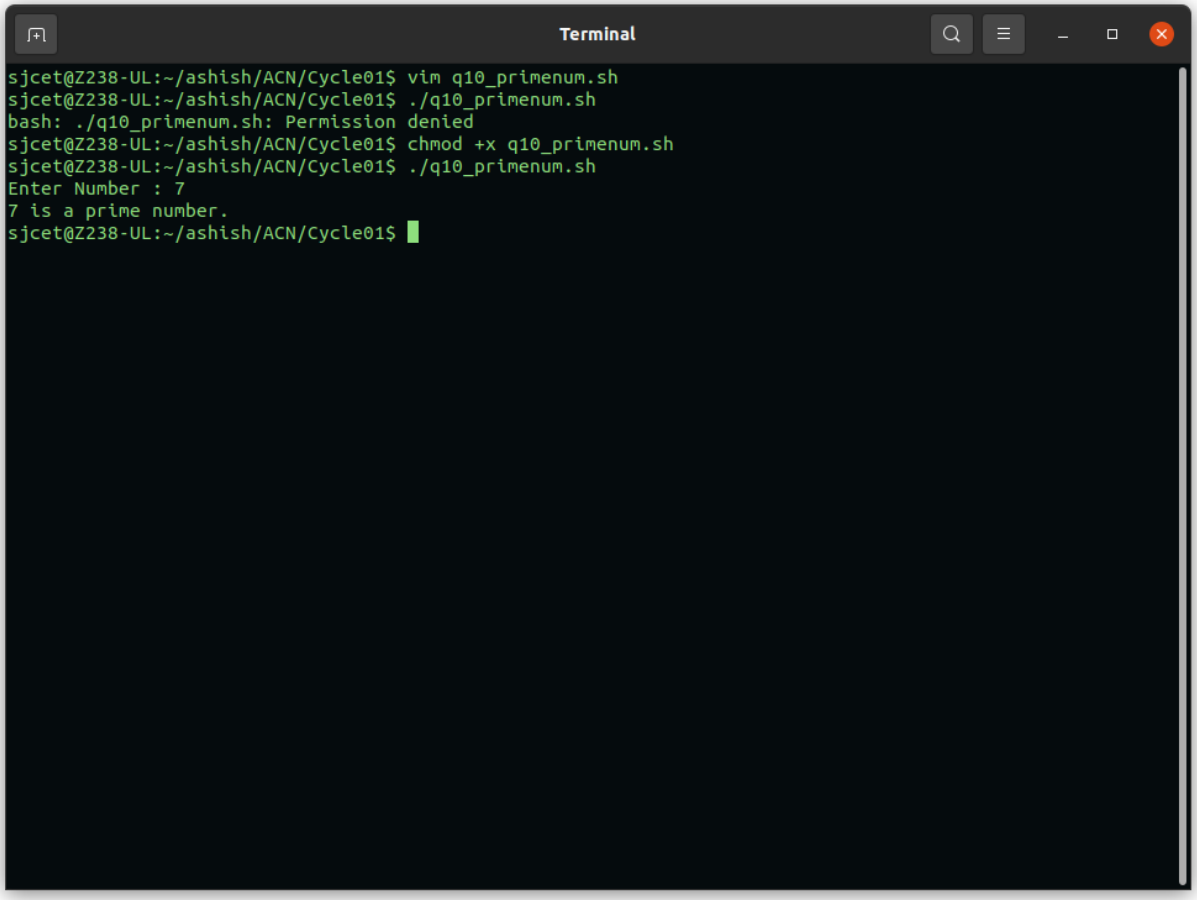
```
echo ”enter the number”
read n
q=$n
a=0
while [ $q – gt 0 ]
do
r= `expr $q % 10 `
q= `expr $q / 10 `
a=`expr $a + $r /* $r /*$r `
done
if [ $a=$n ]
then
echo “the number $n is armstrong number”
else
echo “the number $n is not armstrong number”
fi
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q09_armstrong.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q09_armstrong.sh
”enter the number”
153
./q09_armstrong.sh: line 5: [: too many arguments
“the number 153 is armstrong number”
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

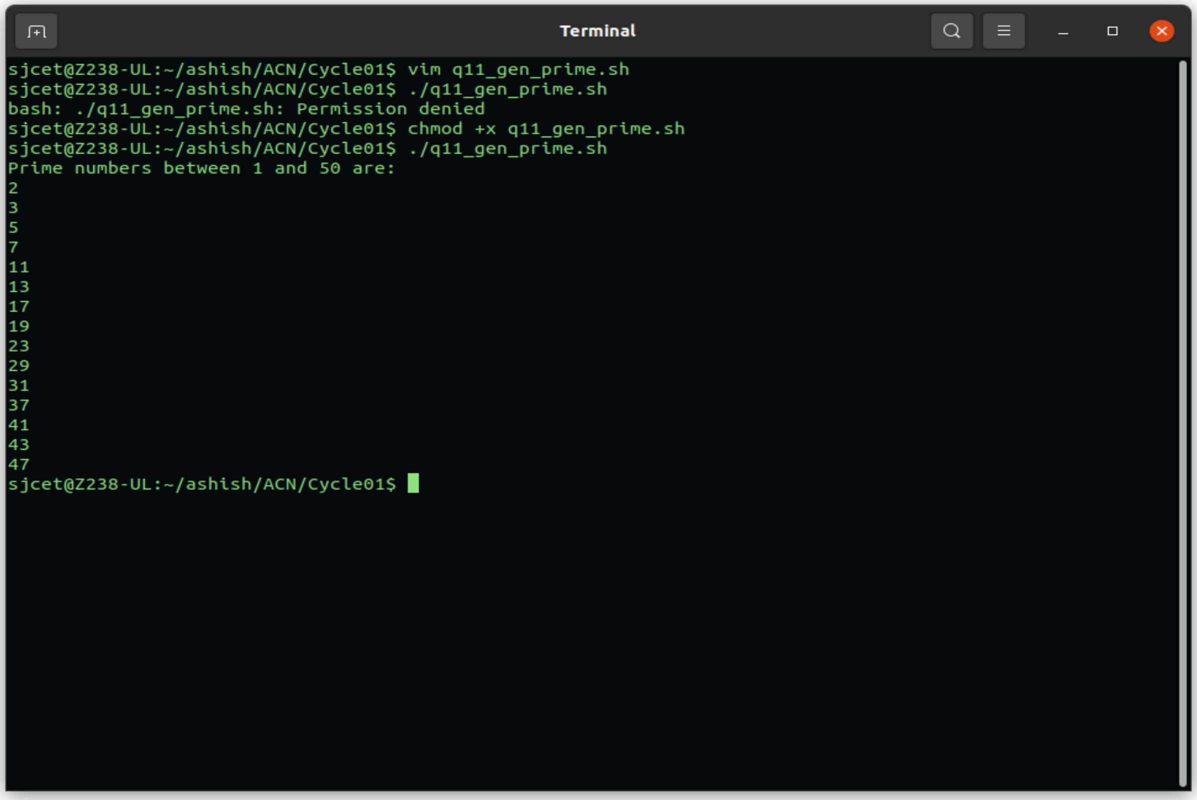
```
echo -e "Enter Number : \c"
read n
for((i=2; i<=$n/2; i++))
do
  ans=$(( n%i ))
  if [ $ans -eq 0 ]
  then
    echo "$n is not a prime number."
    exit 0
  fi
done
echo "$n is a prime number."
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q10_primenum.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q10_primenum.sh
bash: ./q10_primenum.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q10_primenum.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q10_primenum.sh
Enter Number : 7
7 is a prime number.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

```
echo "Prime numbers between 1 and 50 are:"

# Check each number between 1 and 50 for primality
for (( number=2; number<=50; number++ ))
do
    flag=1

    for (( i=2; i<=number/2; i++ ))
    do
        if [ $((number%i)) -eq 0 ]
        then
            flag=0
            break
        fi
    done

    if [ $flag -eq 1 ]
    then
        echo $number
    fi
done
```

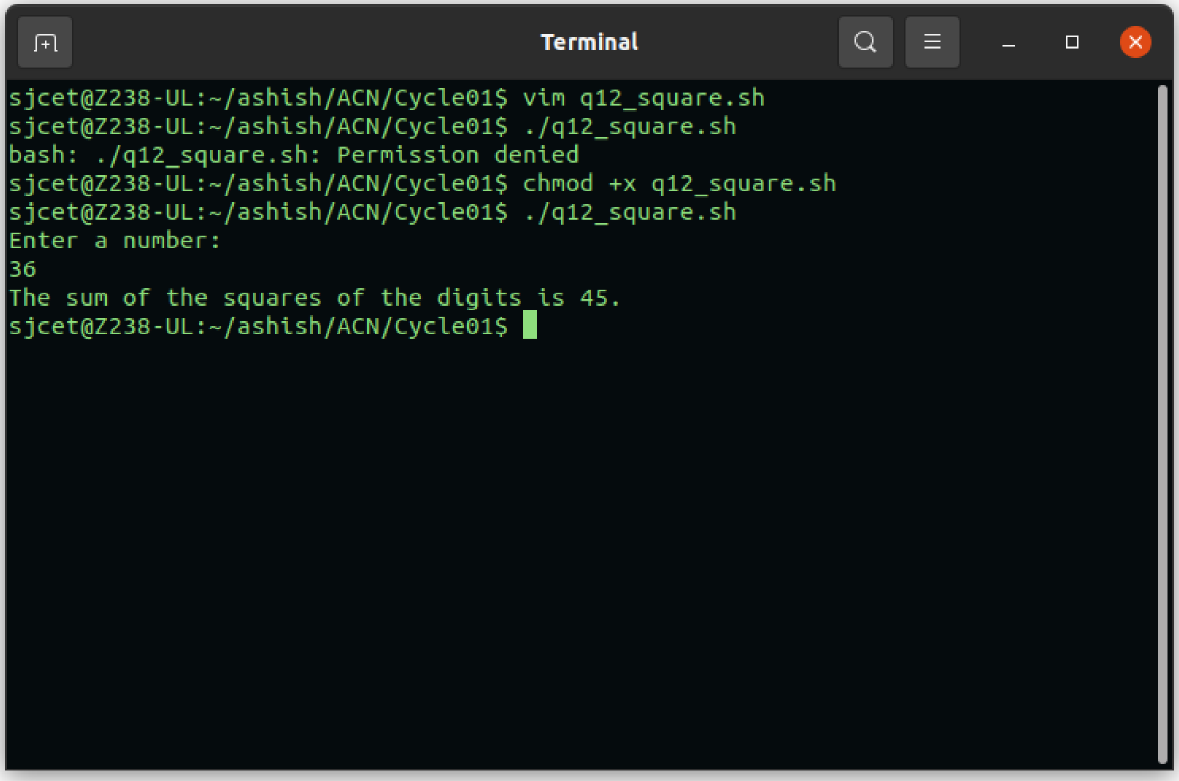


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q11_gen_prime.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q11_gen_prime.sh
bash: ./q11_gen_prime.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q11_gen_prime.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q11_gen_prime.sh
Prime numbers between 1 and 50 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

```
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."

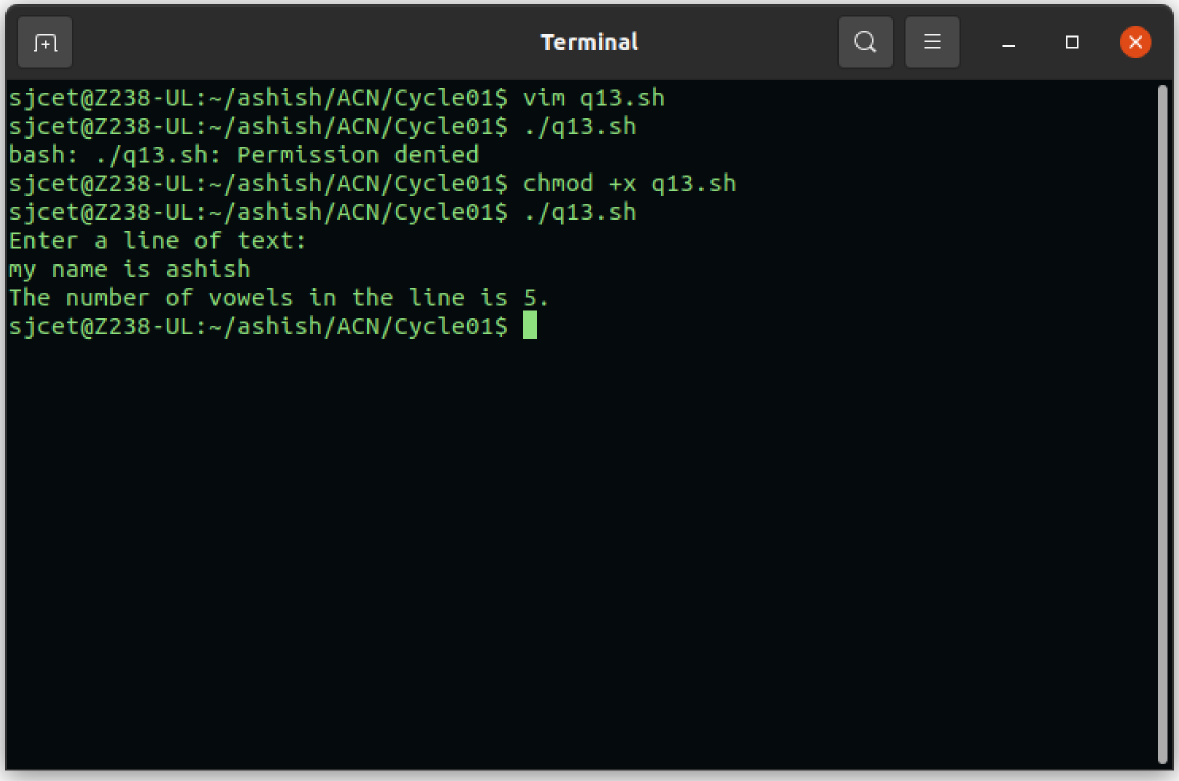


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q12_square.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q12_square.sh
bash: ./q12_square.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q12_square.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q12_square.sh
Enter a number:
36
The sum of the squares of the digits is 45.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

```
echo "Enter a line of text: "
read line

# Initialize the vowel count to 0
count=0

# Loop through each character of the line and check if it is a vowel
for (( i=0; i<${#line}; i++ ))
do
    char=${line:$i:1}
    if [[ $char == [aeiouAEIOU] ]]
    then
        count=$((count + 1))
    fi
done

# Output the result
```

```
echo "The number of vowels in the line is $count."
```

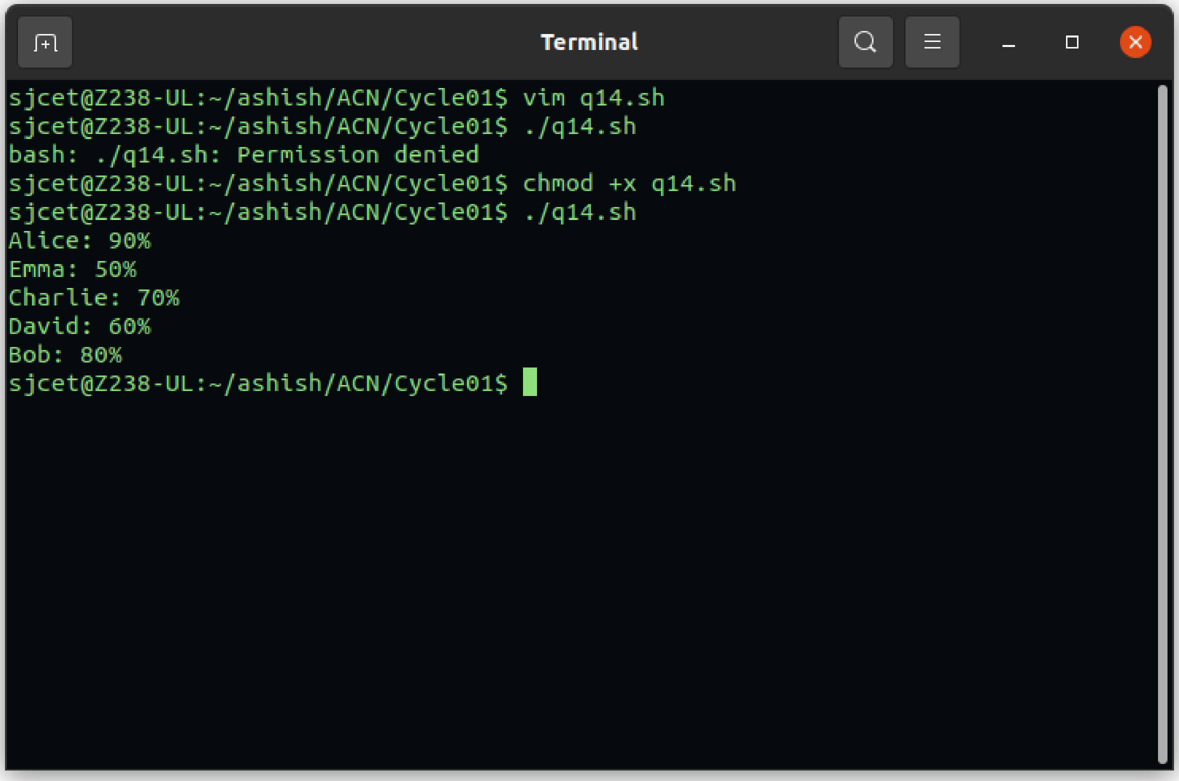


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q13.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q13.sh
bash: ./q13.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q13.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q13.sh
Enter a line of text:
my name is ashish
The number of vowels in the line is 5.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

Computer

14. Write a Shell program to display student grades.

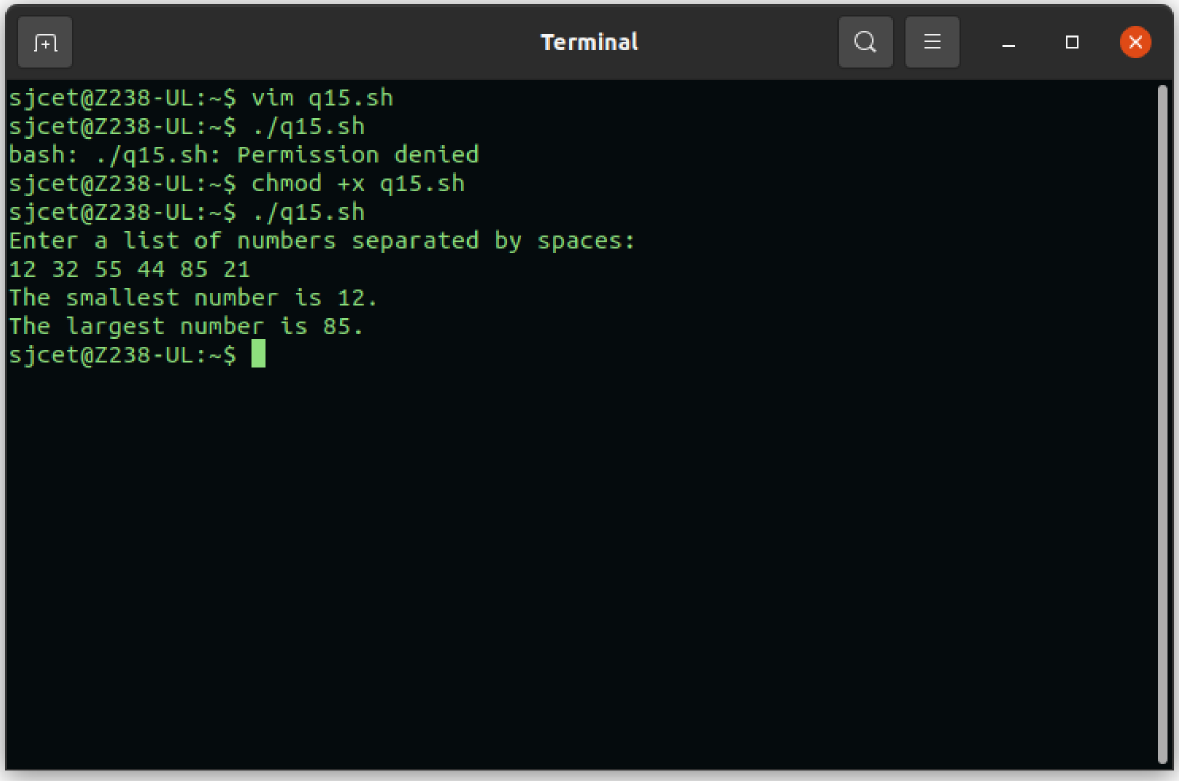
```
declare -A grades=(
    [Alice]=90
    [Bob]=80
    [Charlie]=70
    [David]=60
    [Emma]=50
)

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



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q14.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q14.sh
bash: ./q14.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q14.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q14.sh
Alice: 90%
Emma: 50%
Charlie: 70%
David: 60%
Bob: 80%
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

```
echo "Enter a list of numbers separated by spaces: "
read numbers

# Convert the input string to an array of numbers
IFS=' ' read -ra nums <<< "$numbers"

# Initialize the min and max variables to the first number in the array
min=${nums[0]}
max=${nums[0]}

# Loop through the remaining numbers in the array and update min and max
as needed
for num in "${nums[@]}"
do
    if (( num < min )); then
        min=$num
    fi
    if (( num > max )); then
        max=$num
    fi
done

# Output the result
echo "The smallest number is $min."
echo "The largest number is $max."
```

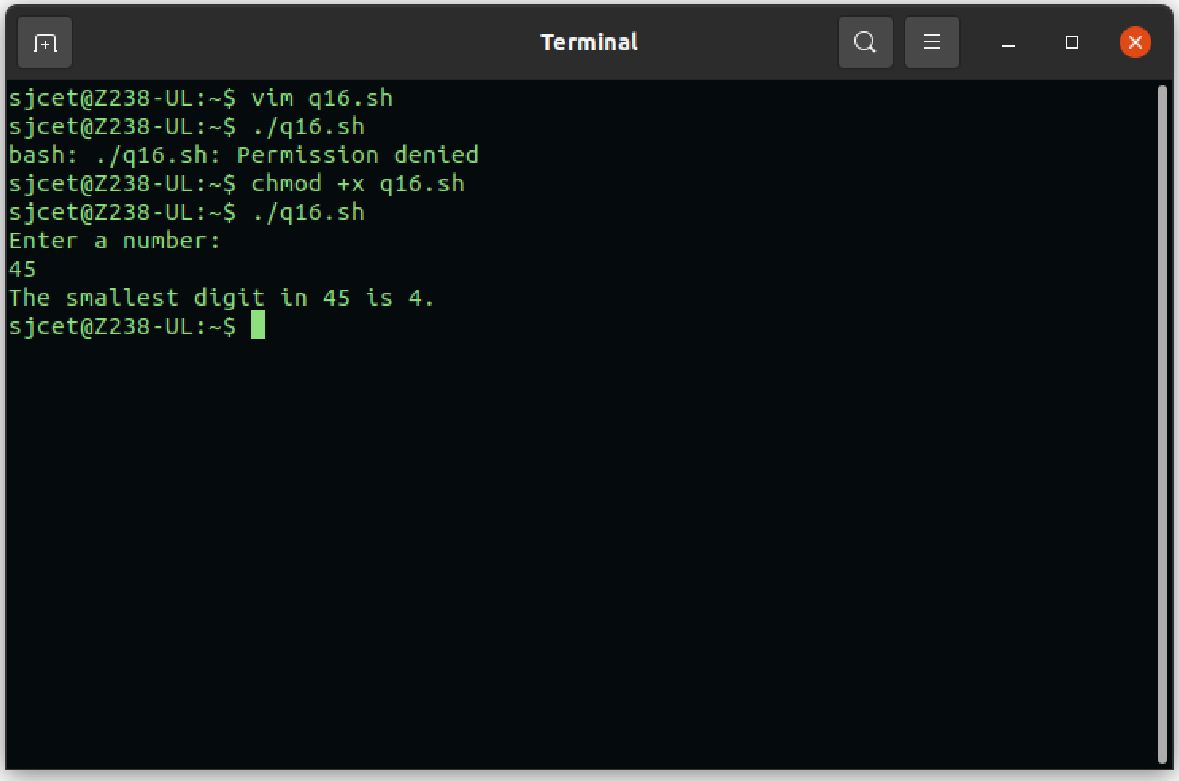


```
sjcet@Z238-UL:~$ vim q15.sh
sjcet@Z238-UL:~$ ./q15.sh
bash: ./q15.sh: Permission denied
sjcet@Z238-UL:~$ chmod +x q15.sh
sjcet@Z238-UL:~$ ./q15.sh
Enter a list of numbers separated by spaces:
12 32 55 44 85 21
The smallest number is 12.
The largest number is 85.
sjcet@Z238-UL:~$ 
```

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

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

# Initialize the min variable to the first digit of the number
min=${num:0:1}

# Loop through the remaining digits of the number and update min as needed
for (( i=1; i<${#num}; i++ ))
do
    digit=${num:$i:1}
    if (( digit < min )); then
```

```
        min=$digit
    fi
done

# Output the result
echo "The smallest digit in $num is $min."
```

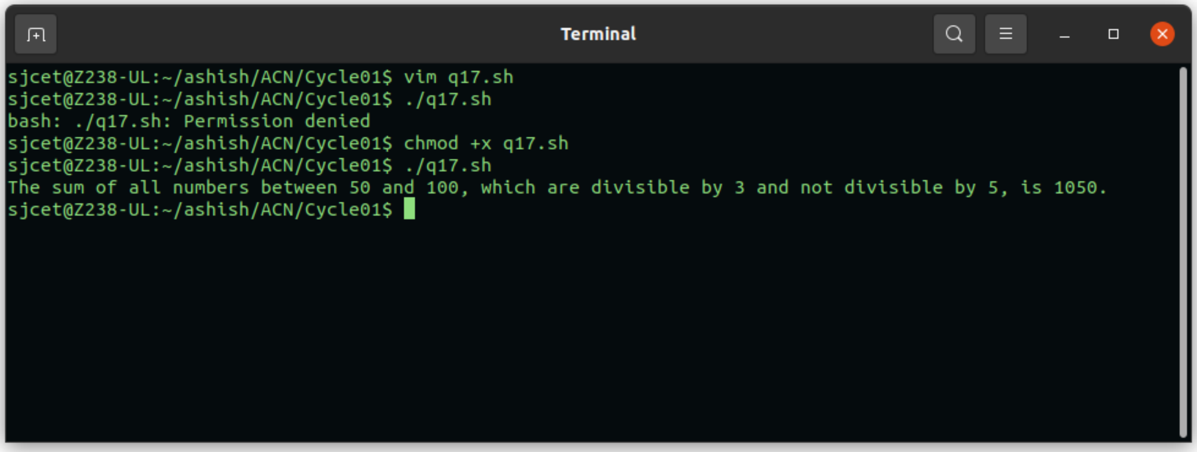


```
sjcet@Z238-UL:~$ vim q16.sh
sjcet@Z238-UL:~$ ./q16.sh
bash: ./q16.sh: Permission denied
sjcet@Z238-UL:~$ chmod +x q16.sh
sjcet@Z238-UL:~$ ./q16.sh
Enter a number:
45
The smallest digit in 45 is 4.
sjcet@Z238-UL:~$
```

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.

```
sum=0

# Loop through the numbers between 50 and 100
for (( num=50; num<=100; num++ ))
do
    # Check if the number is divisible by 3 and not divisible by 5
    if (( num % 3 == 0 && num % 5 != 0 )); then
        sum=$((sum + num))
    fi
done

# Output the result
```

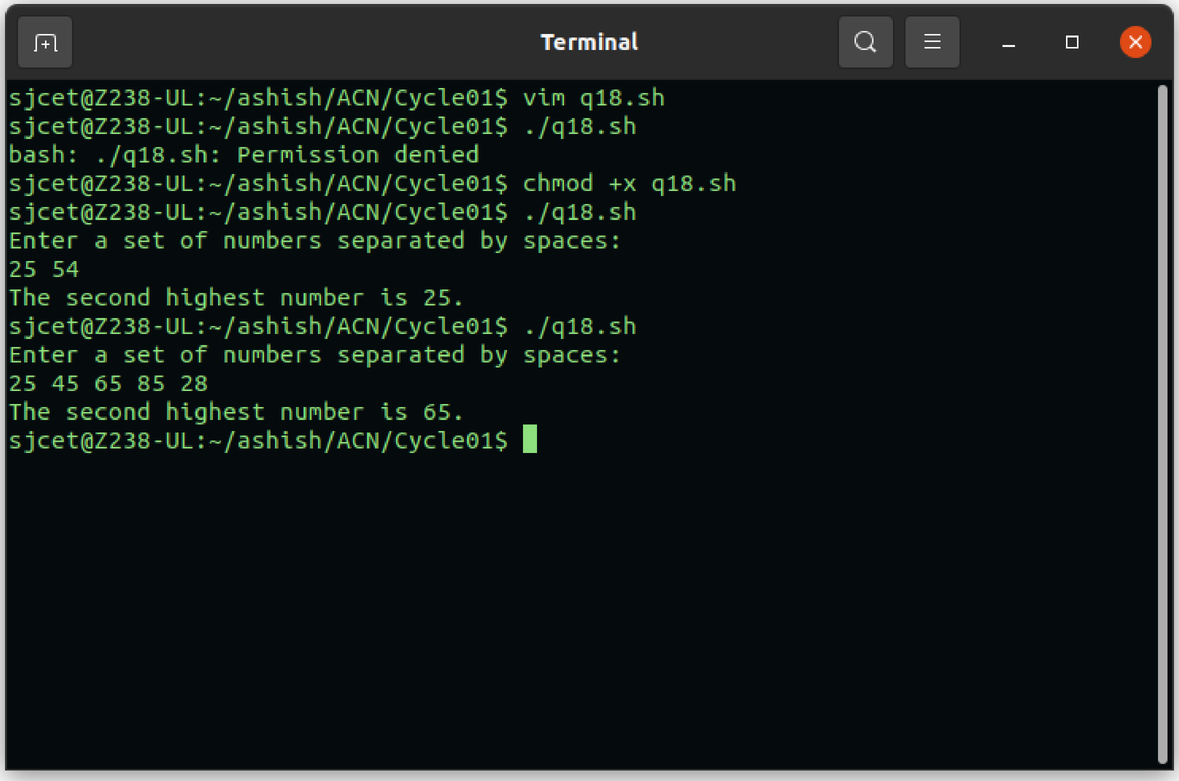
```
echo "The sum of all numbers between 50 and 100, which are divisible by 3
and not divisible by 5, is $sum."
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q17.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q17.sh
bash: ./q17.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q17.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q17.sh
The sum of all numbers between 50 and 100, which are divisible by 3 and not divisible by 5, is 1050.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

```
echo "Enter a set of numbers separated by spaces: "
read numbers

# Convert the space-separated string to an array
arr=($numbers)

# Sort the array in descending order
sorted_arr=($(echo "${arr[@]}" | tr " " "\n" | sort -rn))

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

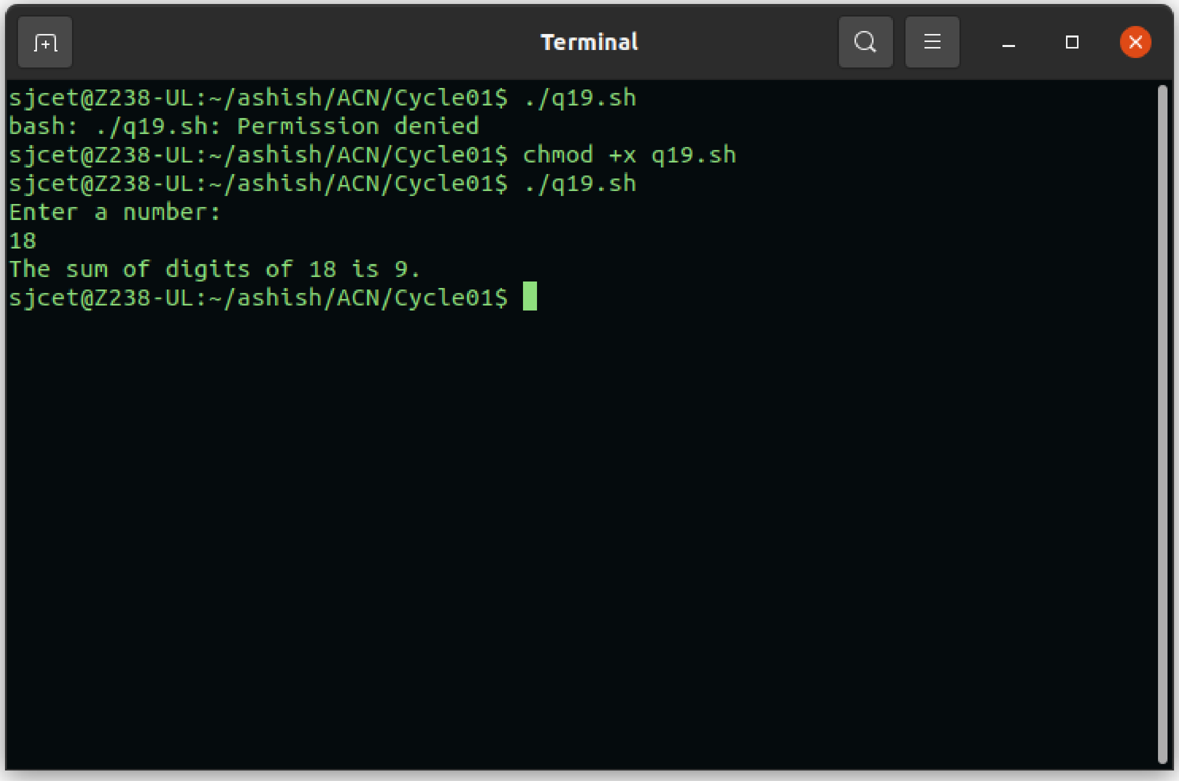


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q18.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q18.sh
bash: ./q18.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q18.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q18.sh
Enter a set of numbers separated by spaces:
25 54
The second highest number is 25.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q18.sh
Enter a set of numbers separated by spaces:
25 45 65 85 28
The second highest number is 65.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

```
# Define the function to calculate the sum of digits
sum_of_digits() {
    num=$1
    sum=0
    while [ $num -gt 0 ]
    do
        digit=$((num % 10))
        sum=$((sum + digit))
        num=$((num / 10))
    done
    echo $sum
}

# Prompt the user to enter a number
echo "Enter a number: "
read num

# Call the function to calculate the sum of digits
result=$(sum_of_digits $num)
```

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

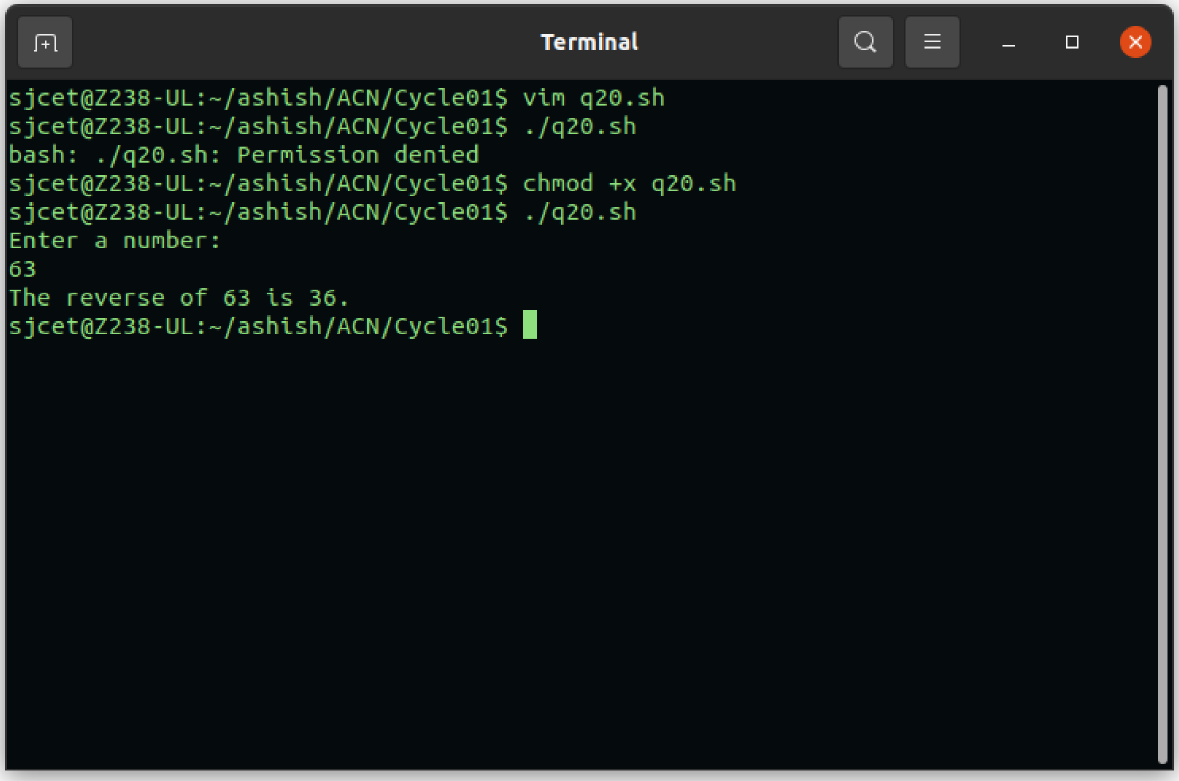


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q19.sh
bash: ./q19.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q19.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q19.sh
Enter a number:
18
The sum of digits of 18 is 9.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

```
# Define the function to reverse a number
reverse_number() {
    num=$1
    rev=0
    while [ $num -gt 0 ]
    do
        digit=$((num % 10))
        rev=$((rev * 10 + digit))
        num=$((num / 10))
    done
    echo $rev
}

# Prompt the user to enter a number
echo "Enter a number: "
read num
```

```
# Call the function to reverse the number
result=$(reverse_number $num)

# Output the result
echo "The reverse of $num is $result."
```

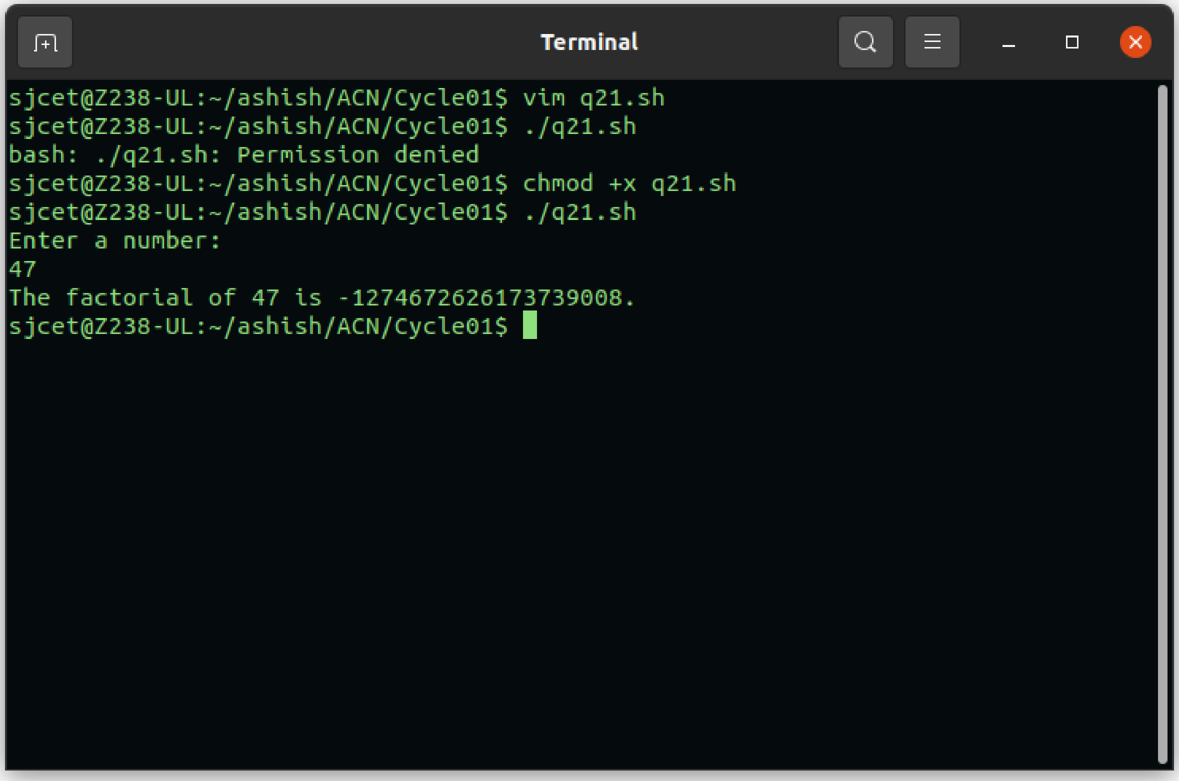


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q20.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q20.sh
bash: ./q20.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q20.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q20.sh
Enter a number:
63
The reverse of 63 is 36.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

```
# Prompt the user to enter a number
echo "Enter a number: "
read num

# Initialize the factorial to 1
factorial=1

# Calculate the factorial using a for loop
for (( i=1; i<=$num; i++ ))
do
    factorial=$((factorial * i))
done
```

```
# Output the result
echo "The factorial of $num is $factorial."
```

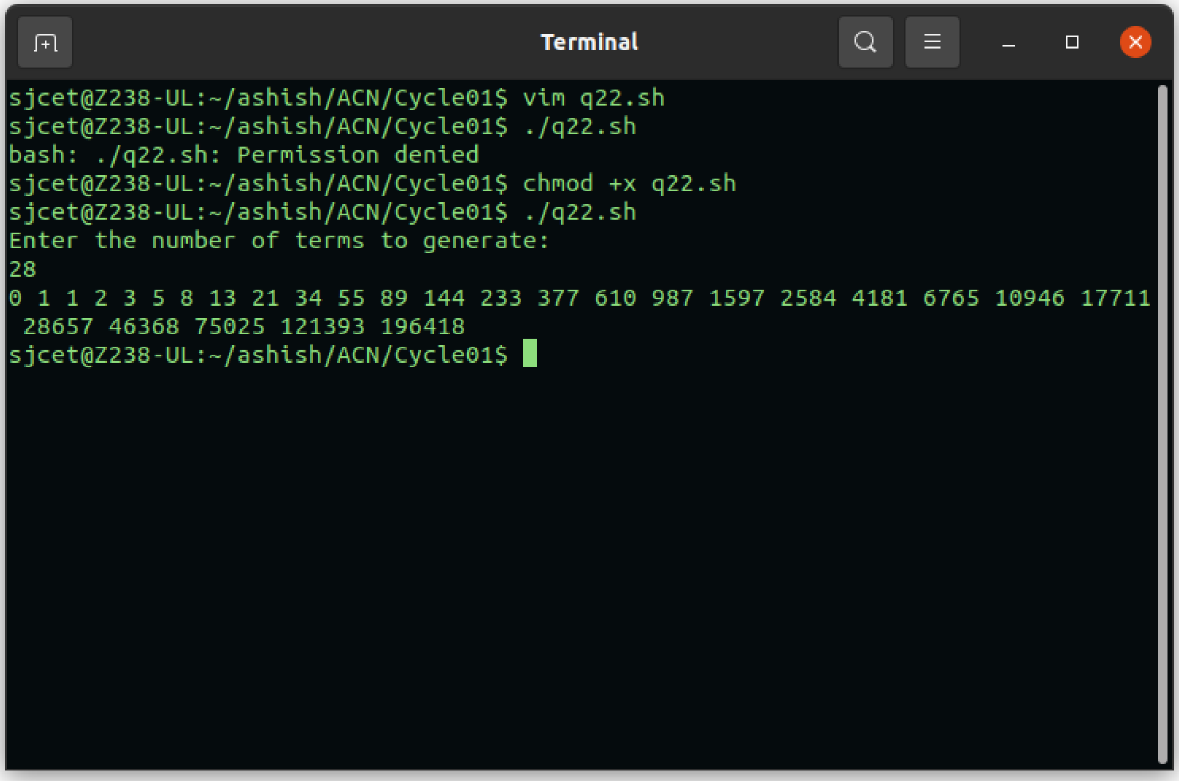


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q21.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q21.sh
bash: ./q21.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q21.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q21.sh
Enter a number:
47
The factorial of 47 is -1274672626173739008.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

22. Write a Shell program to generate Fibonacci series.

```
# Prompt the user to enter the number of terms to generate
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
```

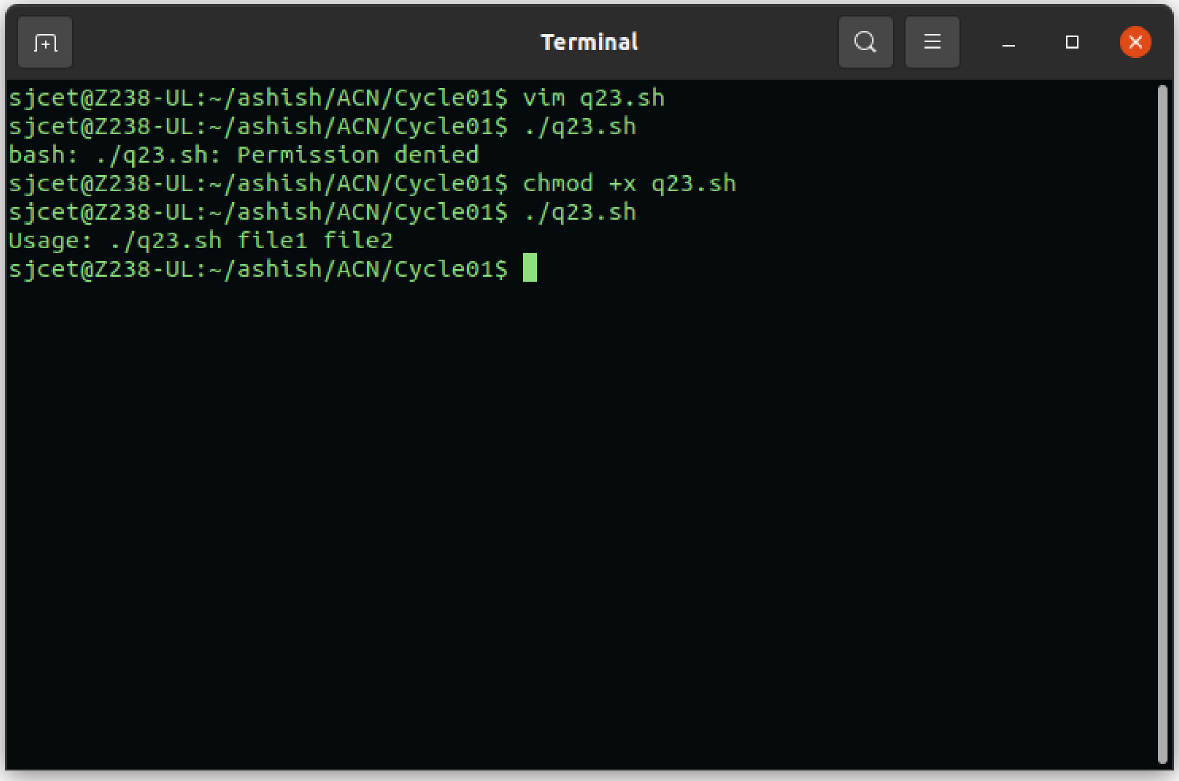


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q22.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q22.sh
bash: ./q22.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q22.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q22.sh
Enter the number of terms to generate:
28
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711
 28657 46368 75025 121393 196418
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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.

```
#!/bin/bash

# check if the correct number of arguments were provided
if [ $# -ne 2 ]; then
  echo "Usage: $0 file1 file2"
  exit 1
```

```
fi

# check if the two files have identical contents
if cmp -s "$1" "$2"; then
  echo "The contents of $1 and $2 are the same."
  rm "$2"
  echo "File $2 has been deleted."
else
  echo "The contents of $1 and $2 are different."
fi
```

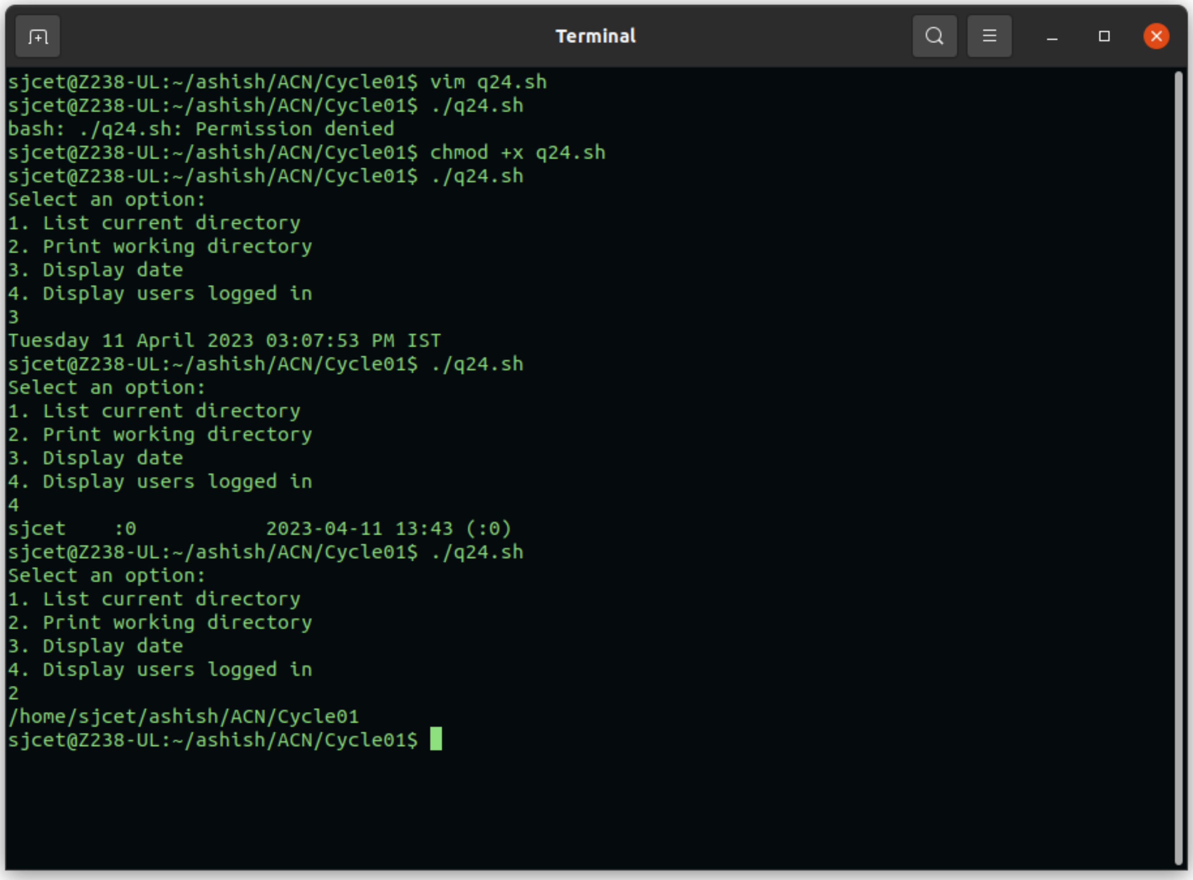


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q23.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q23.sh
bash: ./q23.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q23.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q23.sh
Usage: ./q23.sh file1 file2
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

```
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
```

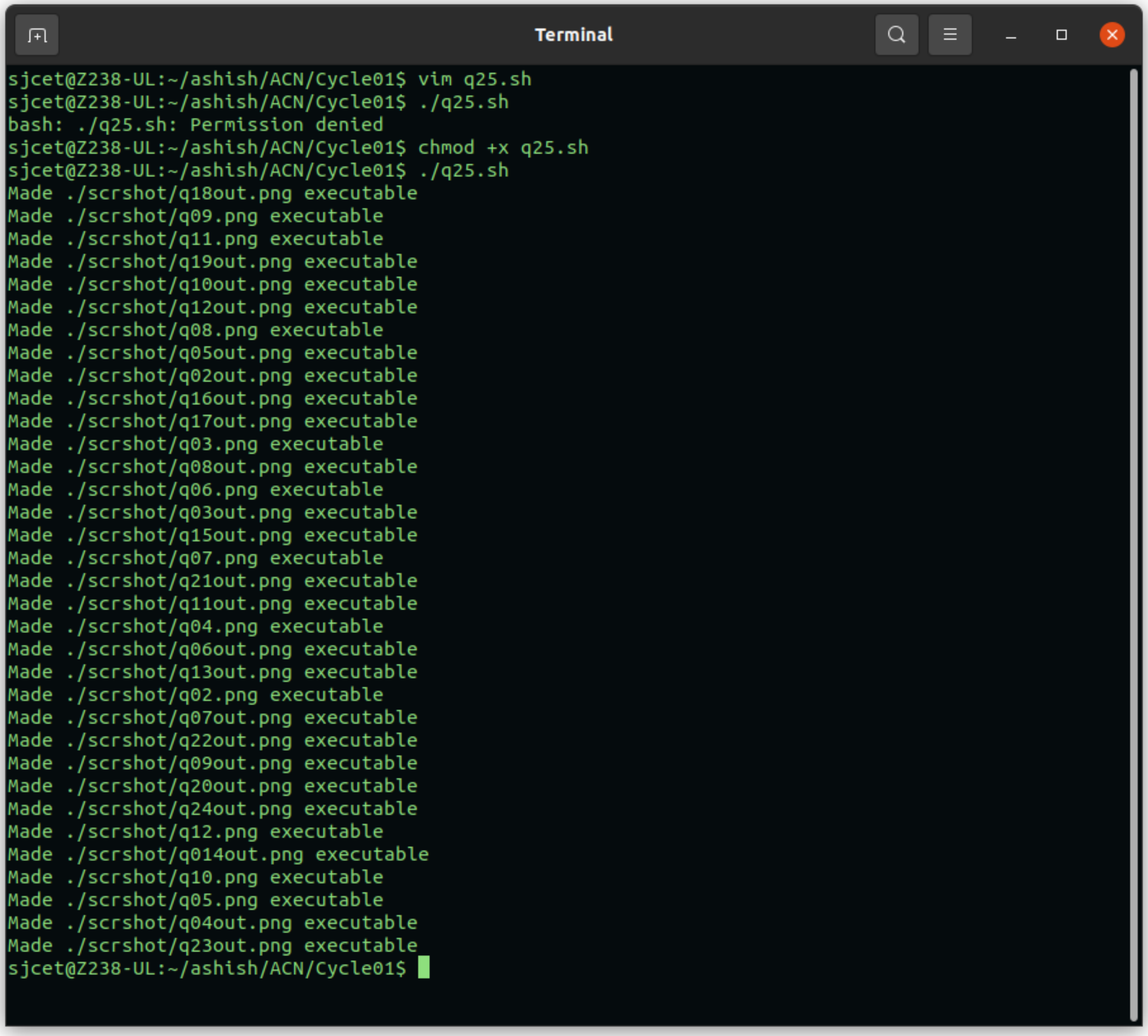
```
case $option in
  1)
    ls -l
    ;;
  2)
    pwd
    ;;
  3)
    date
    ;;
  4)
    who
    ;;
  *)
    echo "Invalid option selected"
    ;;
esac
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q24.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q24.sh
bash: ./q24.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q24.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q24.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 03:07:53 PM IST
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q24.sh
Select an option:
1. List current directory
2. Print working directory
3. Display date
4. Display users logged in
4
sjcet    :0           2023-04-11 13:43 (:0)
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q24.sh
Select an option:
1. List current directory
2. Print working directory
3. Display date
4. Display users logged in
2
/home/sjcet/ashish/ACN/Cycle01
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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.

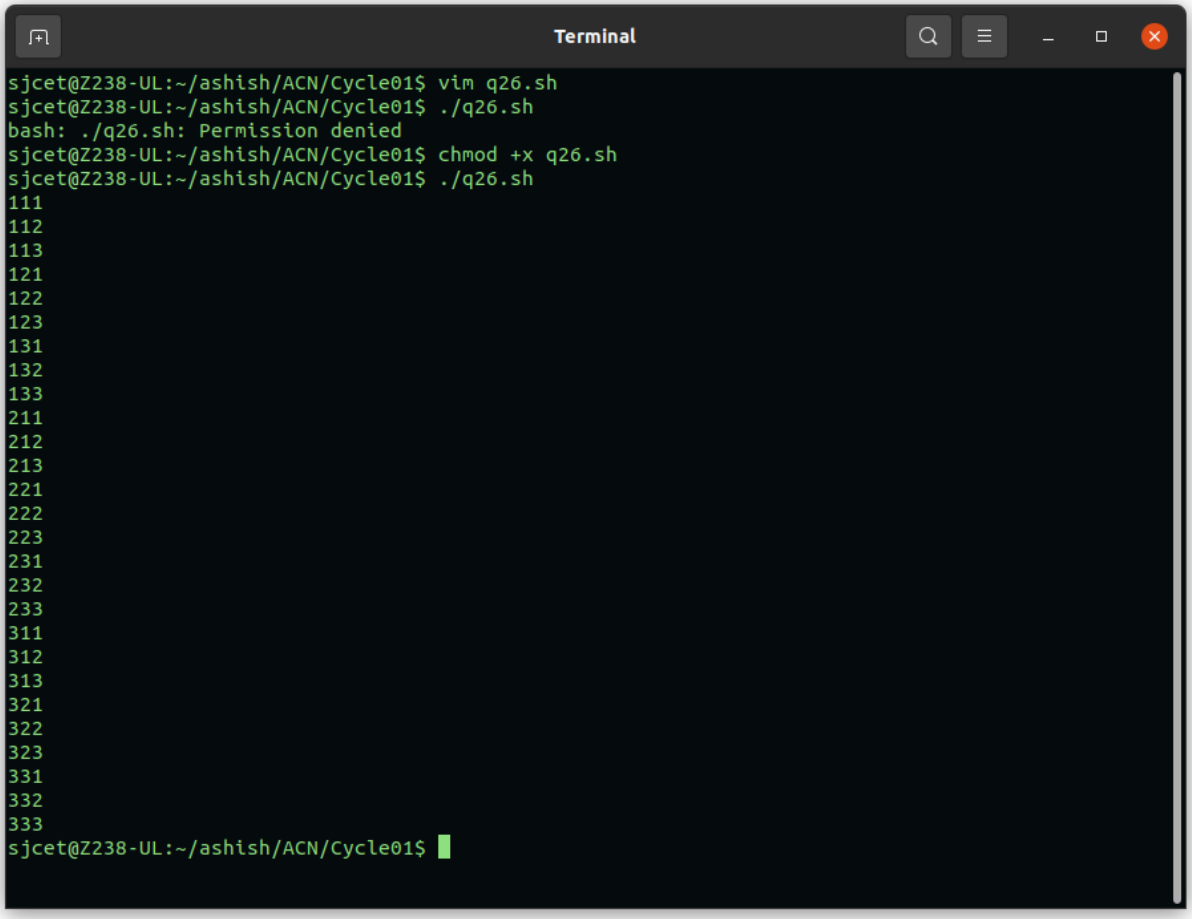
```
find . -type f | while read file; do

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

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



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q25.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q25.sh
bash: ./q25.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q25.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q25.sh
Made ./scrshot/q18out.png executable
Made ./scrshot/q09.png executable
Made ./scrshot/q11.png executable
Made ./scrshot/q19out.png executable
Made ./scrshot/q10out.png executable
Made ./scrshot/q12out.png executable
Made ./scrshot/q08.png executable
Made ./scrshot/q05out.png executable
Made ./scrshot/q02out.png executable
Made ./scrshot/q16out.png executable
Made ./scrshot/q17out.png executable
Made ./scrshot/q03.png executable
Made ./scrshot/q08out.png executable
Made ./scrshot/q06.png executable
Made ./scrshot/q03out.png executable
Made ./scrshot/q15out.png executable
Made ./scrshot/q07.png executable
Made ./scrshot/q21out.png executable
Made ./scrshot/q11out.png executable
Made ./scrshot/q04.png executable
Made ./scrshot/q06out.png executable
Made ./scrshot/q13out.png executable
Made ./scrshot/q02.png executable
Made ./scrshot/q07out.png executable
Made ./scrshot/q22out.png executable
Made ./scrshot/q09out.png executable
Made ./scrshot/q20out.png executable
Made ./scrshot/q24out.png executable
Made ./scrshot/q12.png executable
Made ./scrshot/q014out.png executable
Made ./scrshot/q10.png executable
Made ./scrshot/q05.png executable
Made ./scrshot/q04out.png executable
Made ./scrshot/q23out.png executable
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q26.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q26.sh
bash: ./q26.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q26.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q26.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@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

1

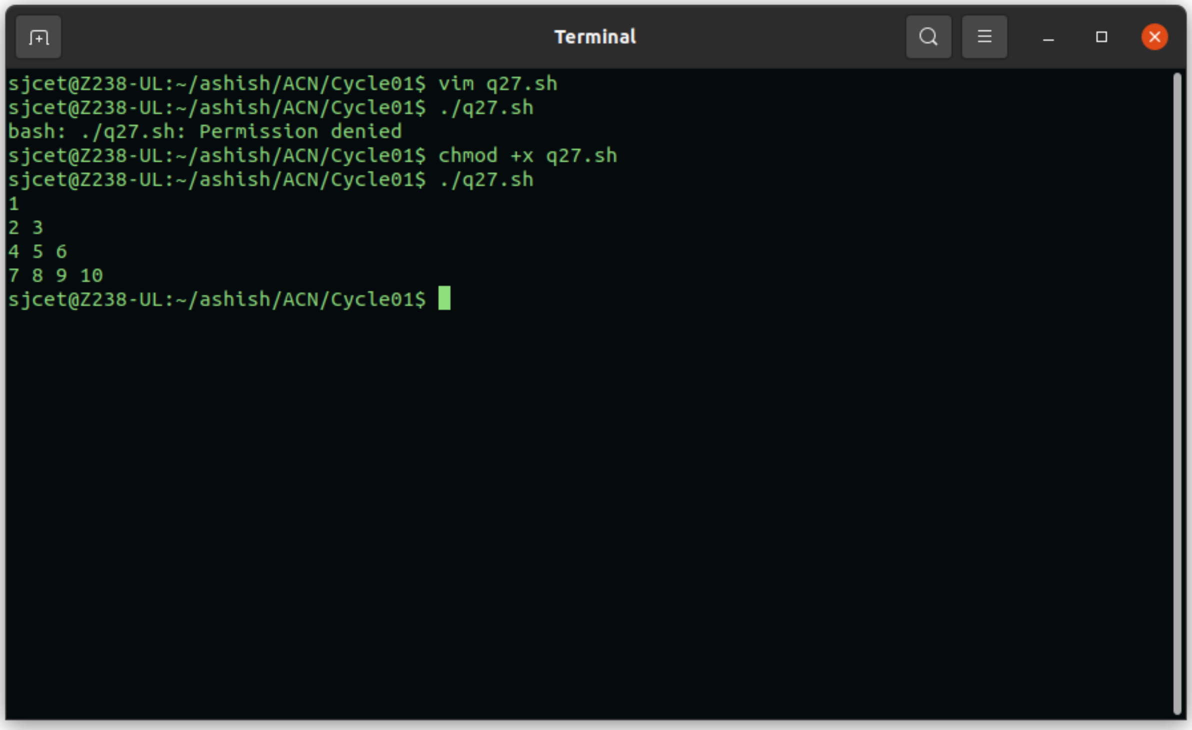
2 3

4 5 6

7 8 9 10

```
count=1

for (( i=1; i<=4; i++ ))
do
```

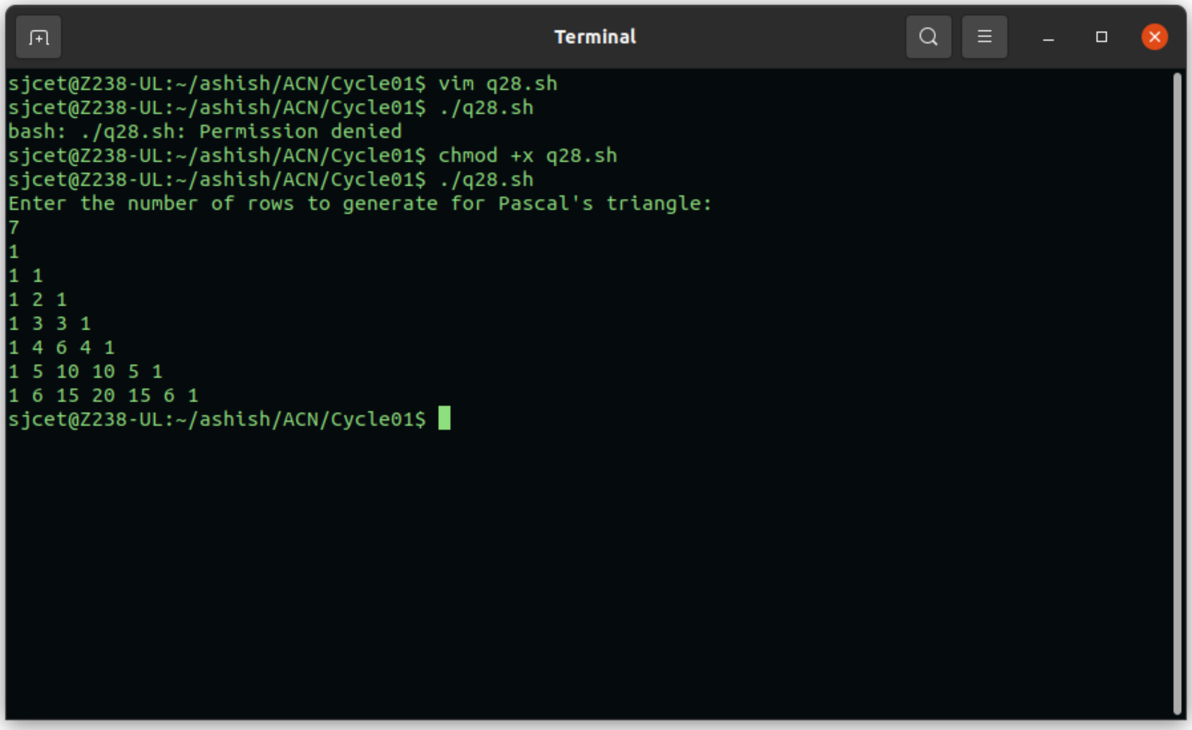
```
  for (( j=1; j<=i; j++ ))
  do
    echo -n "$count "
    count=$((count+1))
  done
  echo ""
done
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q27.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q27.sh
bash: ./q27.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q27.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q27.sh
1
2 3
4 5 6
7 8 9 10
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

```
#!/bin/bash

# set the number of rows
echo "Enter the number of rows to generate for Pascal's triangle:"
read rows

# initialize the first row
row=1
echo $row

# loop over the remaining rows
for ((i=1; i<$rows; i++)); do
  # initialize the row with the left-most element
  prev_row=($row)
  row=${prev_row[0]}

  # loop over the remaining elements in the row
  for ((j=1; j<=i; j++)); do
    # calculate the current element
    current=$((prev_row[j-1] + prev_row[j]))

    # append the current element to the row
    row="$row $current"
  done

  # print the row
  echo $row
done
```

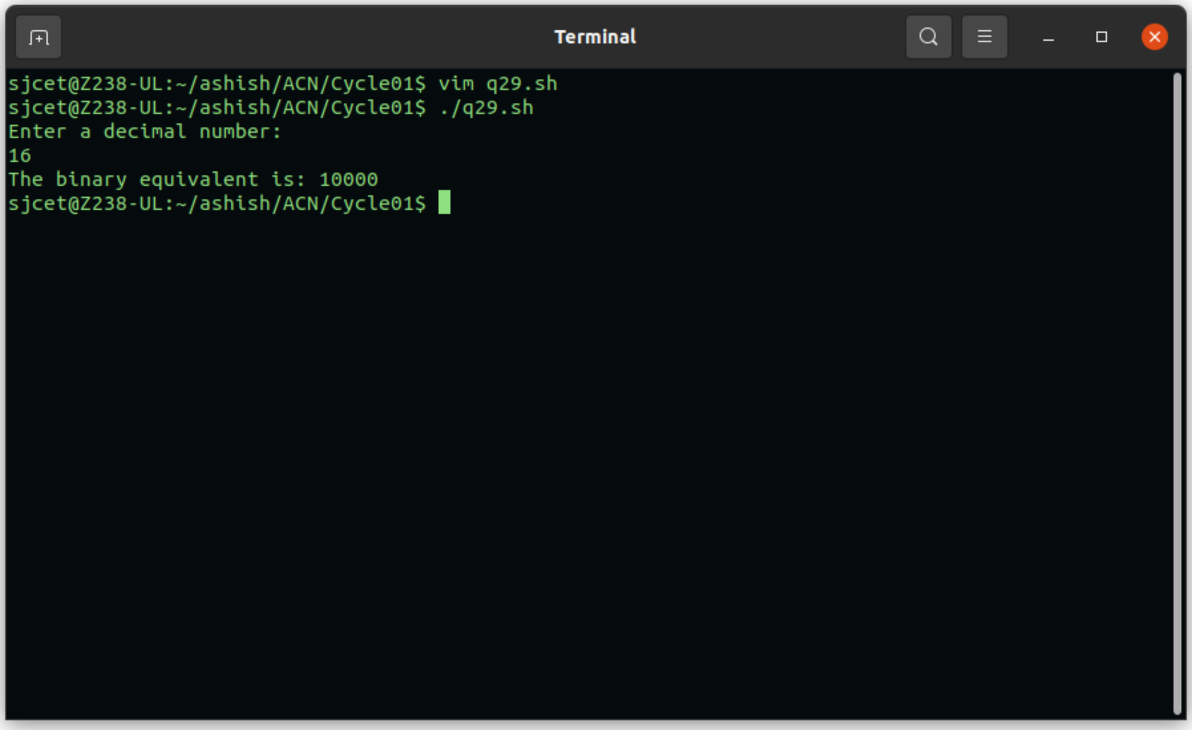


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q28.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q28.sh
bash: ./q28.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q28.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q28.sh
Enter the number of rows to generate for Pascal's triangle:
7
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

29. Write a Decimal to Binary Conversion Shell Script

```
#!/bin/bash

# Prompt the user for the decimal number to convert
echo "Enter a decimal number: "
read decimal
```

```
# Convert the decimal number to binary
binary=""
while [ $decimal -gt 0 ]; do
  remainder=$((decimal % 2))
  binary="$remainder$binary"
  decimal=$((decimal / 2))
done

# Print the binary number
echo "The binary equivalent is: $binary"
```

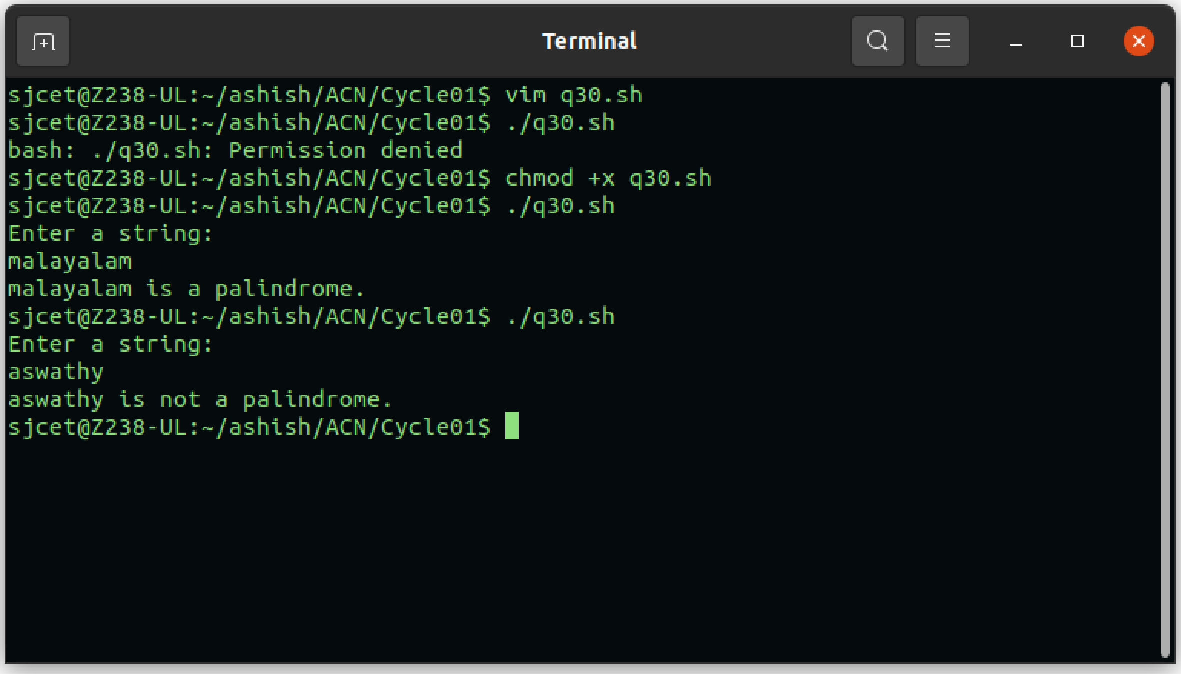


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q29.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q29.sh
Enter a decimal number:
16
The binary equivalent is: 10000
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

```
#!/bin/bash

# Prompt the user for the string to check
echo "Enter a string: "
read string

# Reverse the string
reverse=$(echo $string | rev)

# Check if the string is equal to its reverse
```

```
if [ "$string" == "$reverse" ]; then
  echo "$string is a palindrome."
else
  echo "$string is not a palindrome."
fi
```

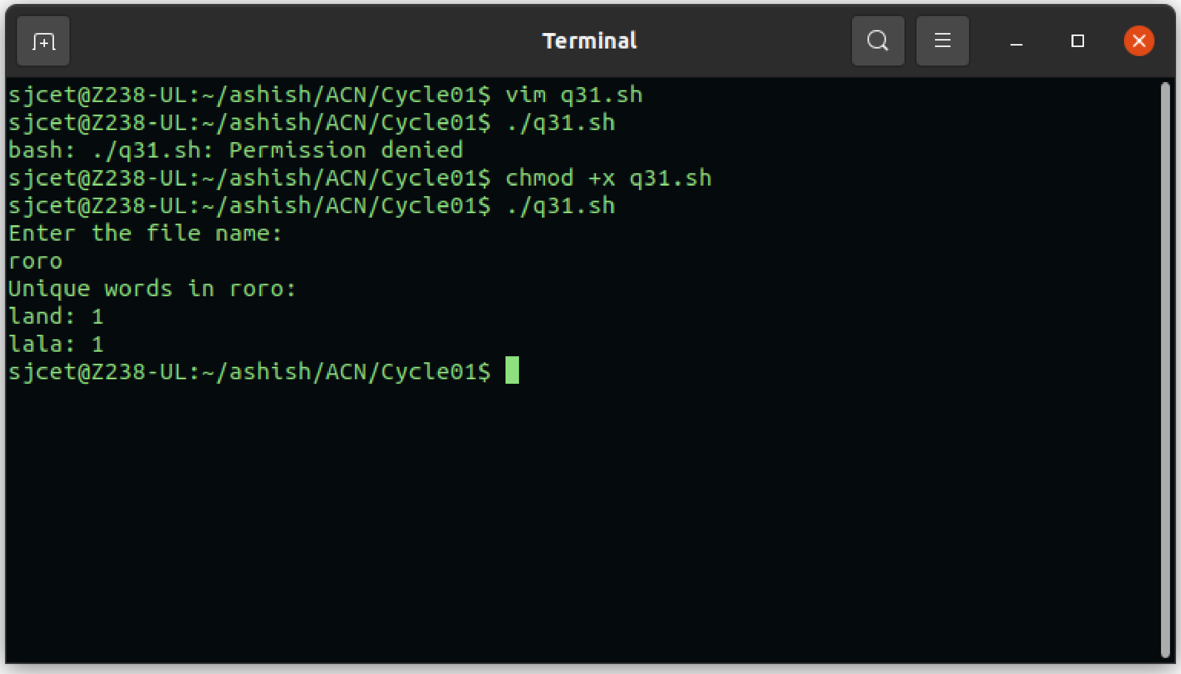


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q30.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q30.sh
bash: ./q30.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q30.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q30.sh
Enter a string:
malayalam
malayalam is a palindrome.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q30.sh
Enter a string:
aswathy
aswathy is not a palindrome.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

```
#!/bin/bash

# Prompt the user for the file name
echo "Enter the file name: "
read file

# Check if the file exists
if [ ! -f "$file" ]; then
  echo "File not found."
  exit 1
fi

# Convert the contents of the file to lowercase and replace all non-
alphanumeric characters with spaces
```

```
contents=$(tr '[:upper:]' '[:lower:]' < $file | sed 's/[^a-z0-9]/ /g')

# Create an array of words from the file contents
words=($contents)

# Loop through the array of words and count their occurrences
declare -A count
for word in "${words[@]}"; do
  if [ -n "$word" ]; then
    ((count[$word]++))
  fi
done

# Print the unique words and their counts
echo "Unique words in $file:"
for word in "${!count[@]}"; do
  echo "$word: ${count[$word]}"
done
```

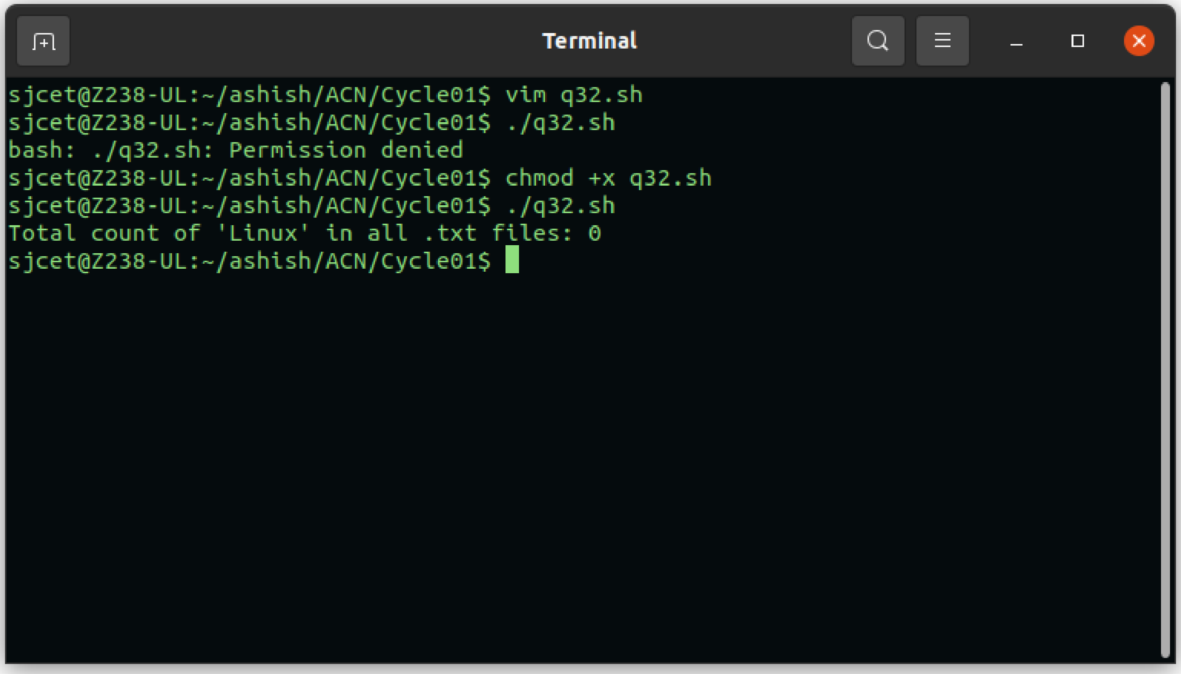


```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q31.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q31.sh
bash: ./q31.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q31.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q31.sh
Enter the file name:
roro
Unique words in roro:
land: 1
lala: 1
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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.

```
#!/bin/bash
```

```
# Set the search directory
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"
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q32.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q32.sh
bash: ./q32.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q32.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q32.sh
Total count of 'Linux' in all .txt files: 0
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

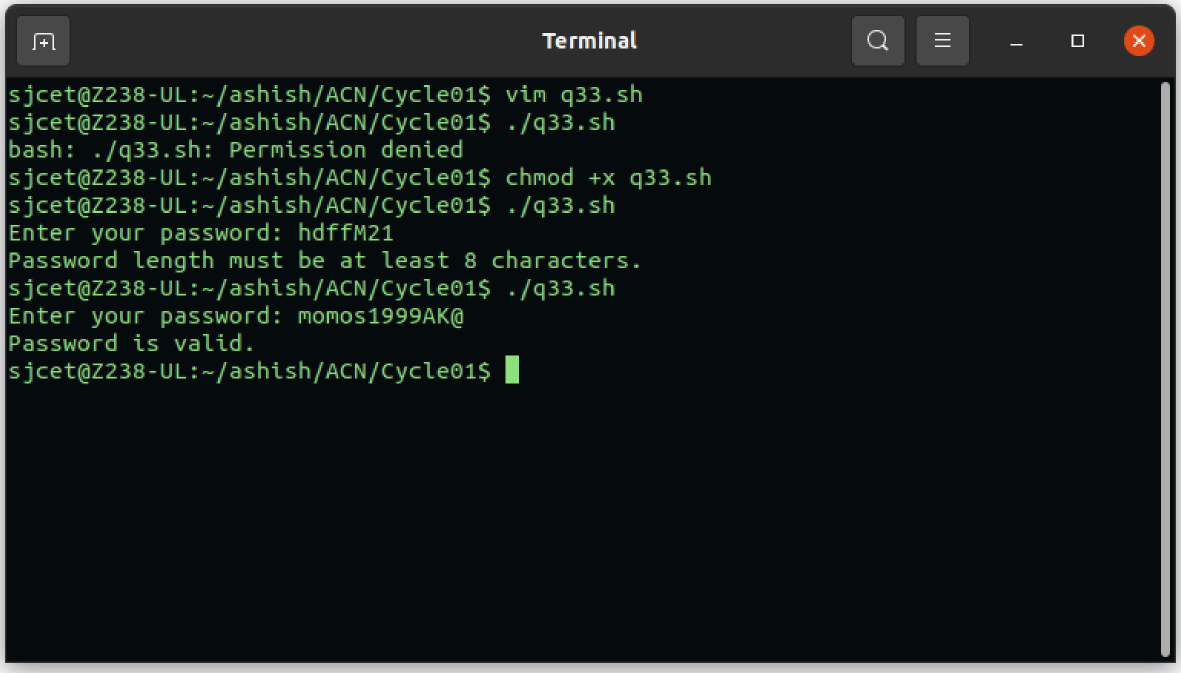
33. Write a shell script to validate password strength. Here are a few assumptions for the password string.

Length – minimum of 8 characters.

Contain both alphabet and number.

Include both the small and capital case letters.

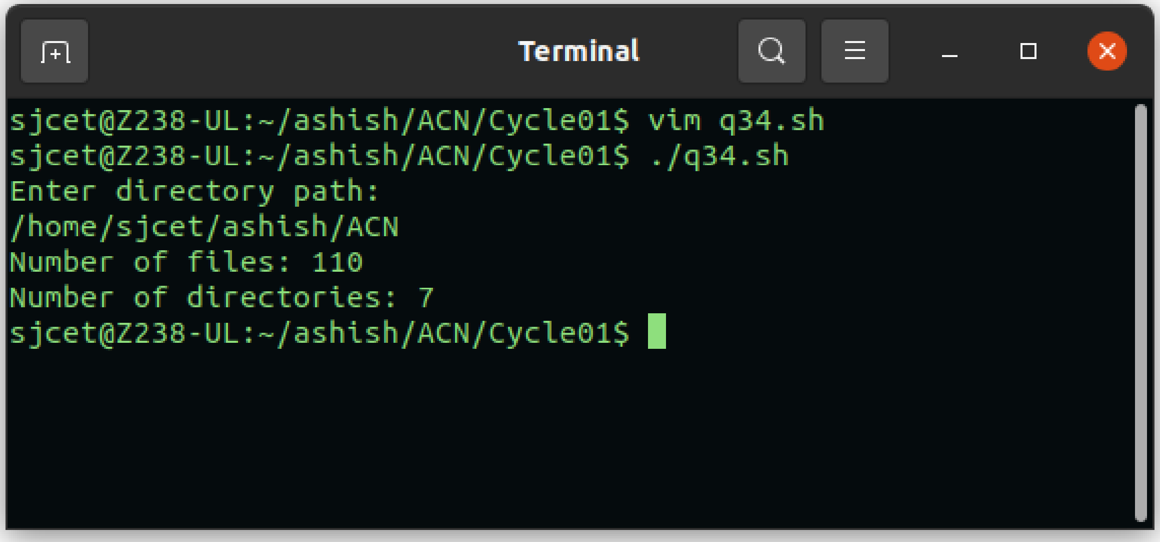
```
#!/bin/bash

read -p "Enter your password: " password

# Check if password is at least 8 characters long
if [[ ${#password} -lt 8 ]]; then
    echo "Password length must be at least 8 characters."
    exit 1
fi

# Check if password contains both alphabet and number
if ! [[ "$password" =~ [A-Za-z]+[0-9]+ ]]; then
    echo "Password must contain both alphabet and number."
    exit 1
fi

# Check if password includes both small and capital case letters
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."
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q33.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q33.sh
bash: ./q33.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q33.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q33.sh
Enter your password: hdffM21
Password length must be at least 8 characters.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q33.sh
Enter your password: momos1999AK@
Password is valid.
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```

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

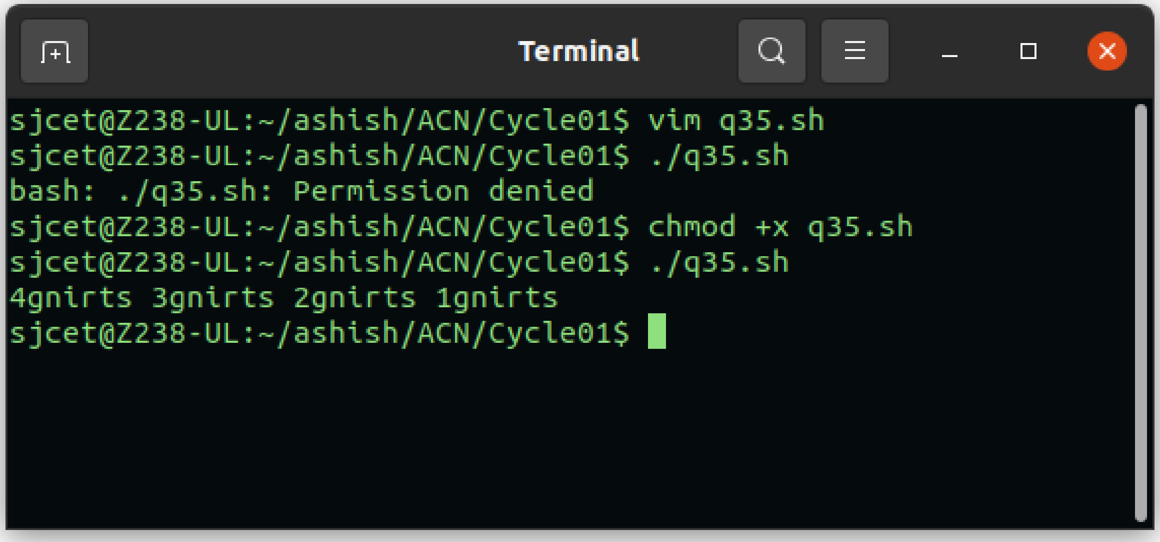
```
echo "Enter directory path: "
read directory

num_files=$(find $directory -type f | wc -l)
num_directories=$(find $directory -type d | wc -l)


echo "Number of files: $num_files"
echo "Number of directories: $num_directories"
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q34.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q34.sh
Enter directory path:
/home/sjcet/ashish/ACN
Number of files: 110
Number of directories: 7
sjcet@Z238-UL:~/ashish/ACN/Cycle01$
```

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

```
#!/bin/bash

# Define a list of strings
my_list=("string1" "string2" "string3" "string4")

# Reverse the order of the list
my_list=($(echo "${my_list[@]}" | tr ' ' '\n' | tac | tr '\n' ' '))

# Reverse each string in the list
for i in "${!my_list[@]}"
do
  my_list[$i]=`echo ${my_list[$i]} | rev`
done

# Print the reversed list of strings
echo "${my_list[@]}"
```



```
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ vim q35.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q35.sh
bash: ./q35.sh: Permission denied
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ chmod +x q35.sh
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ ./q35.sh
4gnirts 3gnirts 2gnirts 1gnirts
sjcet@Z238-UL:~/ashish/ACN/Cycle01$ 
```