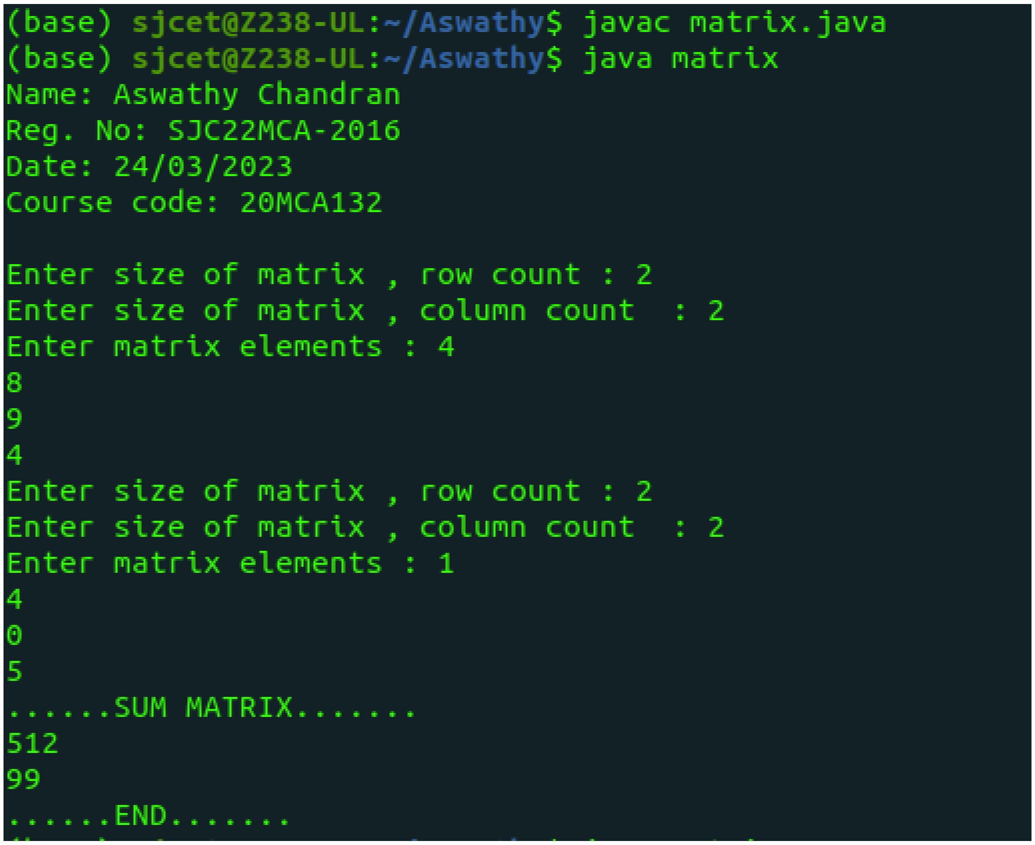
NETWORKING LAB PROGRAMS

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

Code:

```
echo -n "Enter a number:"
read n
if [ `expr $n % 2` == 0 ]
then
        echo "$n is even"
else
        echo "$n is Odd"
fi
```



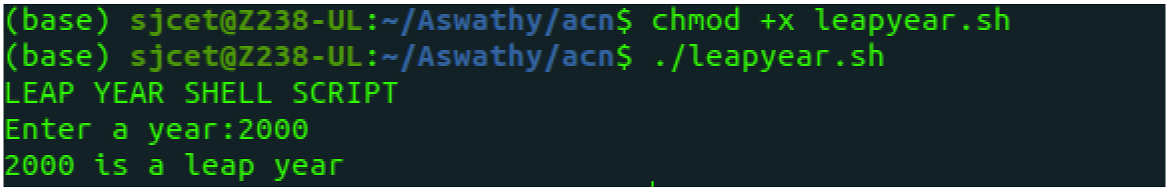
```
(base) sjcet@Z238-UL:~/Aswathy$ javac matrix.java
(base) sjcet@Z238-UL:~/Aswathy$ java matrix
Name: Aswathy Chandran
Reg. No: SJC22MCA-2016
Date: 24/03/2023
Course code: 20MCA132

Enter size of matrix , row count : 2
Enter size of matrix , column count  : 2
Enter matrix elements : 4
8
9
4
Enter size of matrix , row count : 2
Enter size of matrix , column count  : 2
Enter matrix elements : 1
4
0
5
......SUM MATRIX.......
512
99
......END.......
```

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

Code:

```
echo "Enter the year (YYYY)"
```

```
read year

if [ $((year % 4)) -eq 0 ]
then
  if [ $((year % 100)) -eq 0 ]
    then
    if [ $((year % 400)) -eq 0 ]
          then
        echo "$year is a leap year"
    else
           echo "$year is not a leap year"
    fi
  else
  echo "$year is a leap year"
  fi
else
 echo "$year is not a leap year"

fi
```

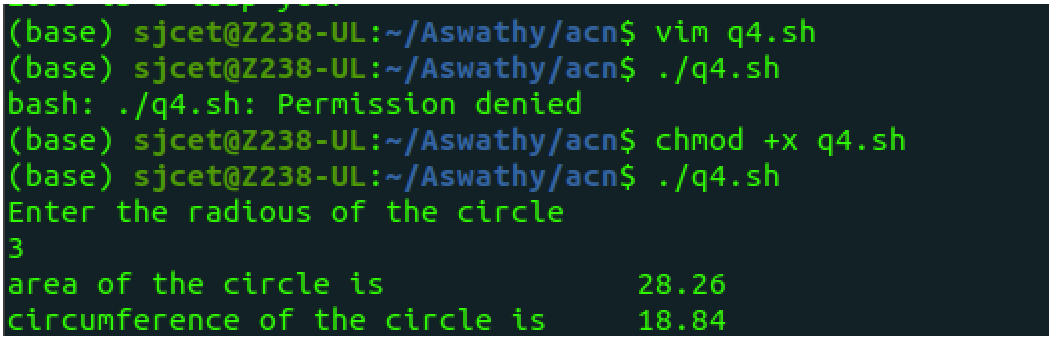


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x leapyear.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./leapyear.sh
LEAP YEAR SHELL SCRIPT
Enter a year:2000
2000 is a leap year
```

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

<u>Code:</u>

```
echo "Enter the radius:"
read r

area=`echo 3.14 \* $r \* $r| bc `
cir=`echo 2 \* 3.14 \* $r| bc `
echo "Area : $area"

echo "Circumference : $cir"
```

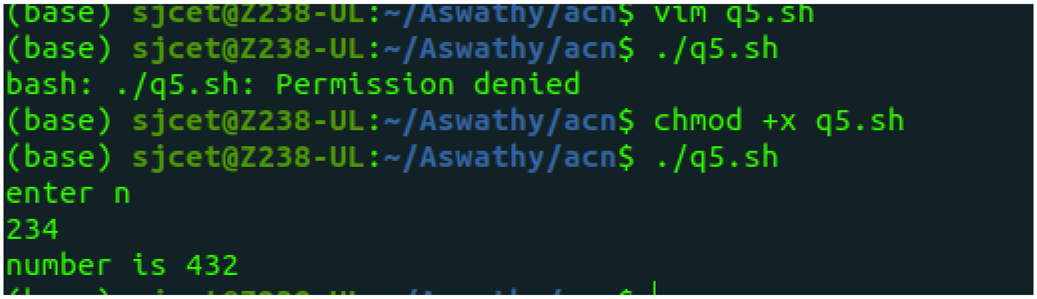


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q4.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q4.sh
bash: ./q4.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q4.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q4.sh
Enter the radious of the circle
3
area of the circle is              28.26
circumference of the circle is     18.84
```

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

Code:

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

reverse=$(echo "$num" | rev)

if [ "$num" -eq "$reverse" ]; then
  echo "$num is same when reversed."
else
  echo "$num is not same when reversed."
fi
```



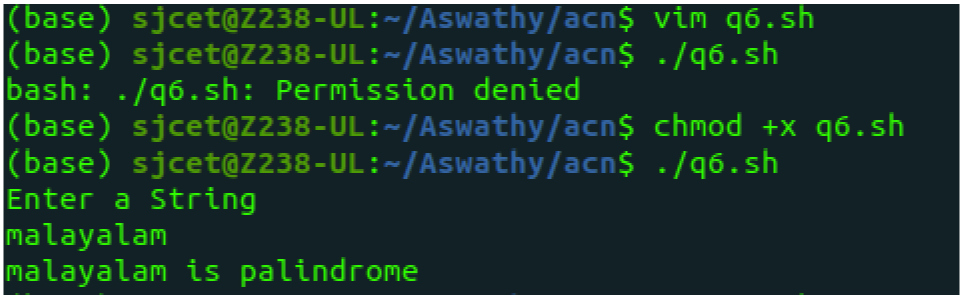
```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q5.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q5.sh
bash: ./q5.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q5.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q5.sh
enter n
234
number is 432
```

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

Code:

```
echo Enter the string
read s
echo $s>temp
rvs="$(rev temp)"
if [ $s = $rvs ]
then
```

```
echo "it is palindrome"
else
echo " it is not a Palindrome"
fi
```

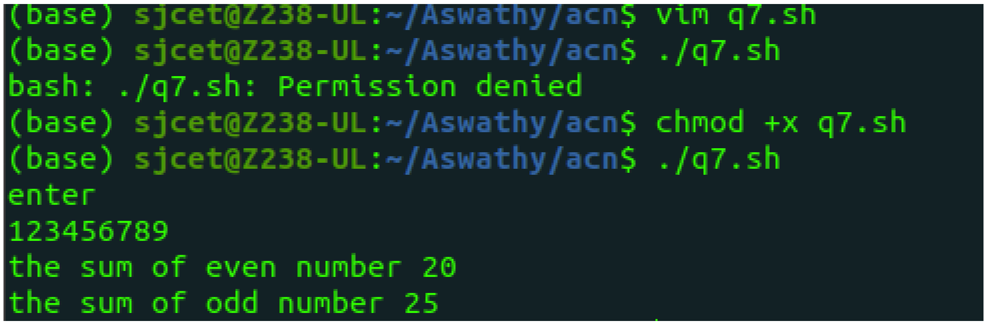


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q6.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q6.sh
bash: ./q6.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q6.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q6.sh
Enter a String
malayalam
malayalam is palindrome
```

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

Code:

```
echo "Enter a set of numbers (separated by spaces):"
read numbers
IFS=" " read -a array <<< "$numbers"
sum_even=0
sum_odd=0
for num in "${array[@]}"
do
   if [ $((num % 2)) -eq 0 ]
   then
      sum_even=$((sum_even + num))
   else
      sum_odd=$((sum_odd + num))
   fi
done

echo "Sum of even numbers: $sum_even"
echo "Sum of odd numbers: $sum_odd"
```

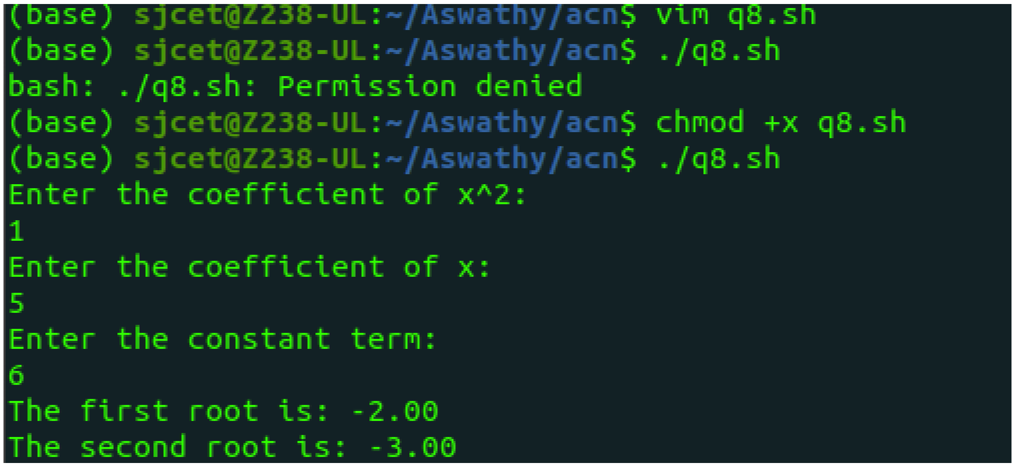


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q7.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q7.sh
bash: ./q7.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q7.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q7.sh
enter
123456789
the sum of even number 20
the sum of odd number 25
```

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

Code:

```
echo "Enter the coefficients of the quadratic equation (a, b, c): "
read a b c


discriminant=$((b*b - 4*a*c))

if [ $discriminant -lt 0 ]
then
   echo "The quadratic equation has no real roots."
else

   root1=$(echo "scale=2; (-$b + sqrt($discriminant)) / (2*$a)" | bc)
   root2=$(echo "scale=2; (-$b - sqrt($discriminant)) / (2*$a)" | bc)

   echo "The roots of the quadratic equation are: $root1 and $root2"
fi
```



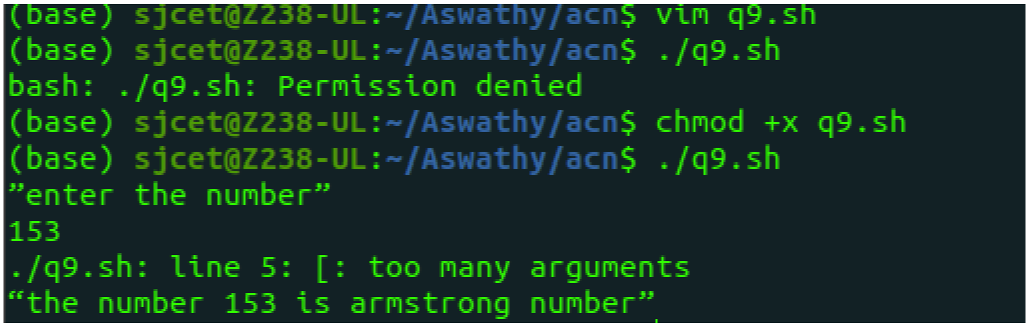
```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q8.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q8.sh
bash: ./q8.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q8.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q8.sh
Enter the coefficient of x^2:
1
Enter the coefficient of x:
5
Enter the constant term:
6
The first root is: -2.00
The second root is: -3.00
```

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

Code:

```
echo "Enter an integer: "
read number


count=${#number}
```

```
sum=0

for (( i=0; i<count; i++ ))
do
   digit=${number:i:1}
   sum=$((sum + digit**count))
done


if [ "$sum" -eq "$number" ]
then
   echo "The number $number is an Armstrong number."
else
   echo "The number $number is not an Armstrong number."
Fi
```



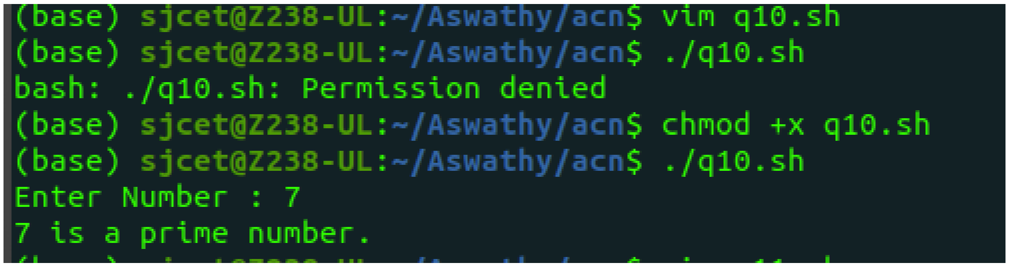
```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q9.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q9.sh
bash: ./q9.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q9.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q9.sh
”enter the number”
153
./q9.sh: line 5: [: too many arguments
“the number 153 is armstrong number”
```

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

Code:

```
echo "Enter an integer: "
read number

flag=1

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

```
done

if [ $number -eq 1 ]
then
   echo "1 is neither prime nor composite."
elif [ $flag -eq 1 ]
then
   echo "$number is a prime number."
else
   echo "$number is not a prime number."
fi
```

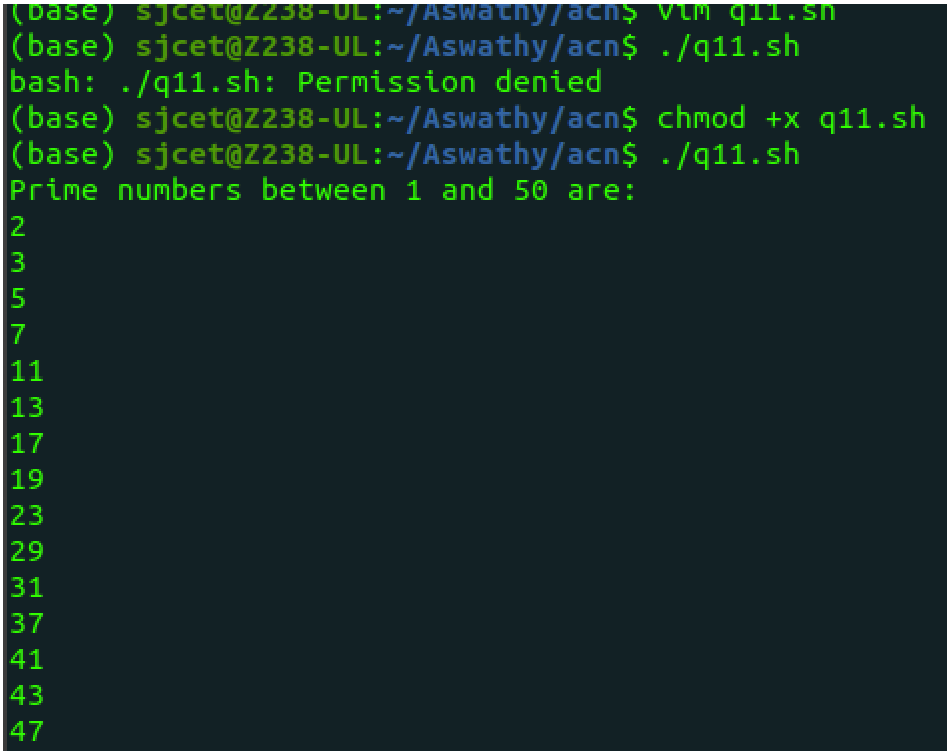


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q10.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q10.sh
bash: ./q10.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q10.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q10.sh
Enter Number : 7
7 is a prime number.
```

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

Code:

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

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
```

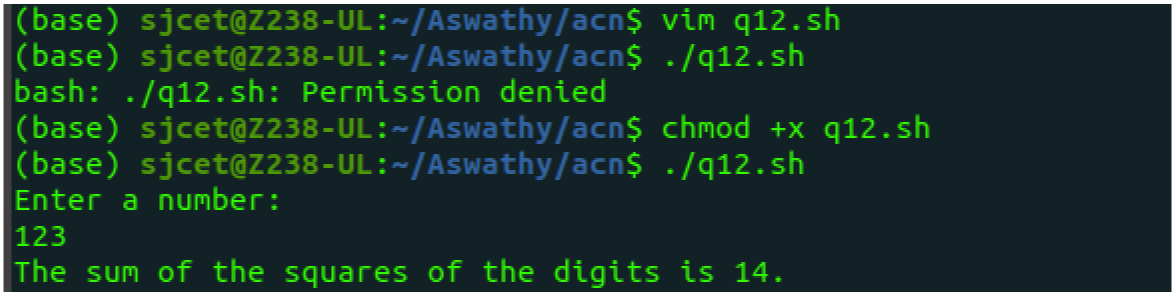


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q11.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q11.sh
bash: ./q11.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q11.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q11.sh
Prime numbers between 1 and 50 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
```

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

Code:

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

sum=0

while [ $number -ne 0 ]
do
   digit=$((number % 10))
   sum=$((sum + digit * digit))
   number=$((number / 10))
done

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

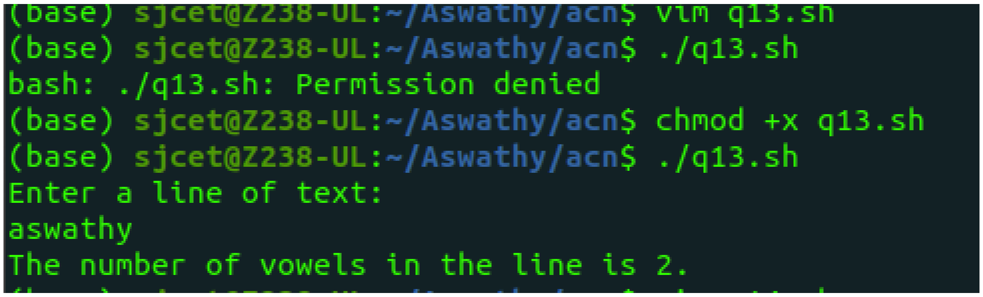


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q12.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q12.sh
bash: ./q12.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q12.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q12.sh
Enter a number:
123
The sum of the squares of the digits is 14.
```

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

Code:

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

count=0

for (( i=0; i<${#line}; i++ ))
do
    char=${line:$i:1}
    if [[ $char == [aeiouAEIOU] ]]
    then
        count=$((count + 1))
    fi
done

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

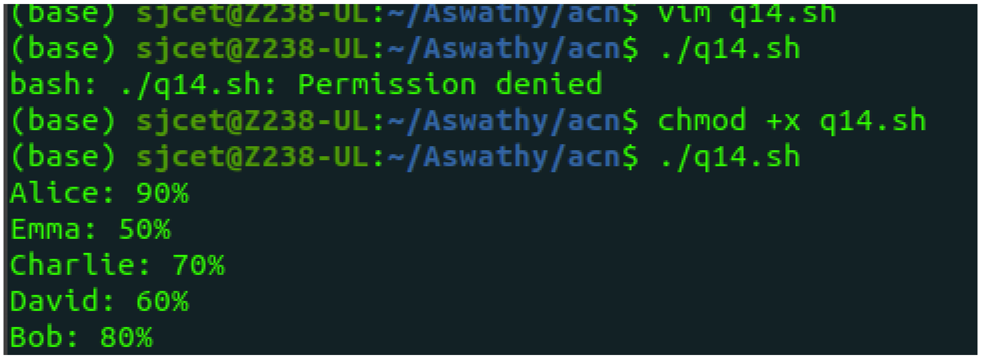


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q13.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q13.sh
bash: ./q13.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q13.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q13.sh
Enter a line of text:
aswathy
The number of vowels in the line is 2.
```

14. Write a Shell program to display student grades.

Code:

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

for name in "${!grades[@]}"
do
   echo "$name: ${grades[$name]}%"
done
```



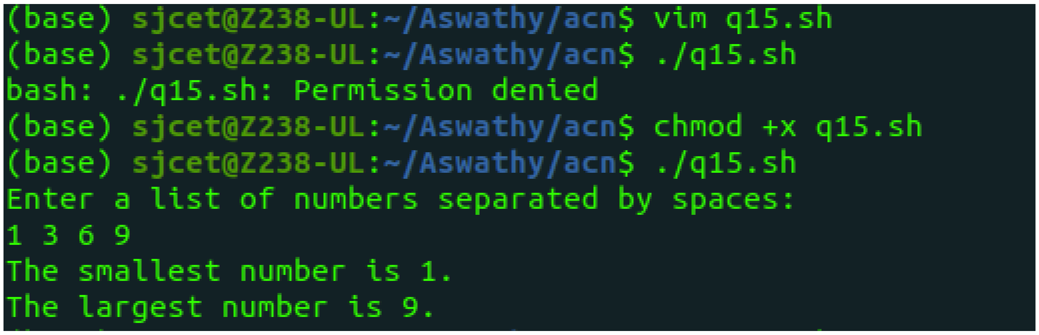
```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q14.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q14.sh
bash: ./q14.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q14.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q14.sh
Alice: 90%
Emma: 50%
Charlie: 70%
David: 60%
Bob: 80%
```

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

Code:

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


IFS=' ' read -ra nums <<< "$numbers"


min=${nums[0]}
max=${nums[0]}

for num in "${nums[@]}"
do
   if (( num < min )); then
      min=$num
   fi
```

```
    if (( num > max )); then
        max=$num
    fi
done


echo "The smallest number is $min."
echo "The largest number is $max."
```

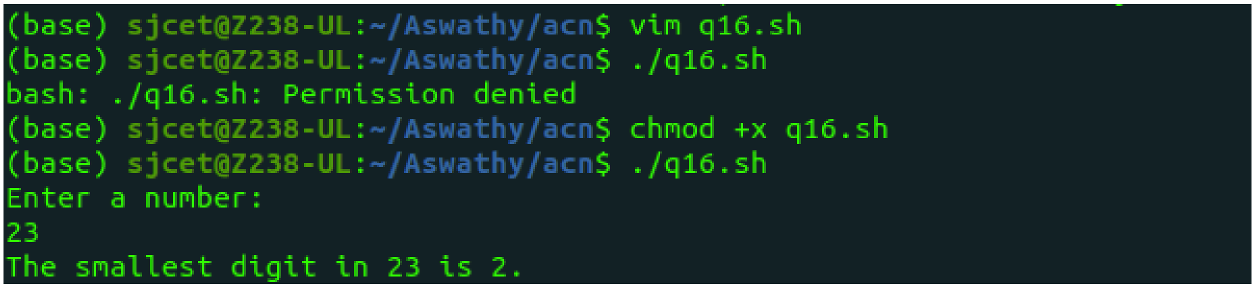


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q15.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q15.sh
bash: ./q15.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q15.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q15.sh
Enter a list of numbers separated by spaces:
1 3 6 9
The smallest number is 1.
The largest number is 9.
```

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

Code:

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


min=${num:0:1}

for (( i=1; i<${#num}; i++ ))
do
    digit=${num:$i:1}
    if (( digit < min )); then
        min=$digit
    fi
done


echo "The smallest digit in $num is $min."
```

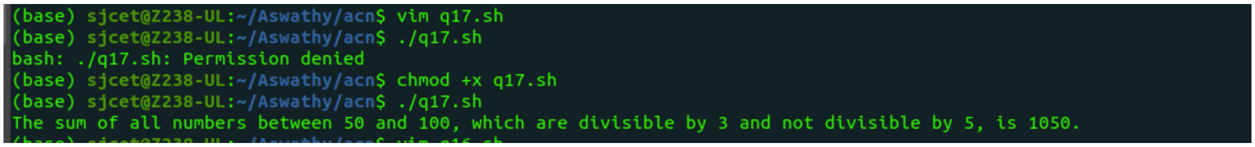


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q16.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q16.sh
bash: ./q16.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q16.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q16.sh
Enter a number:
23
The smallest digit in 23 is 2.
```

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>Code:</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."
```



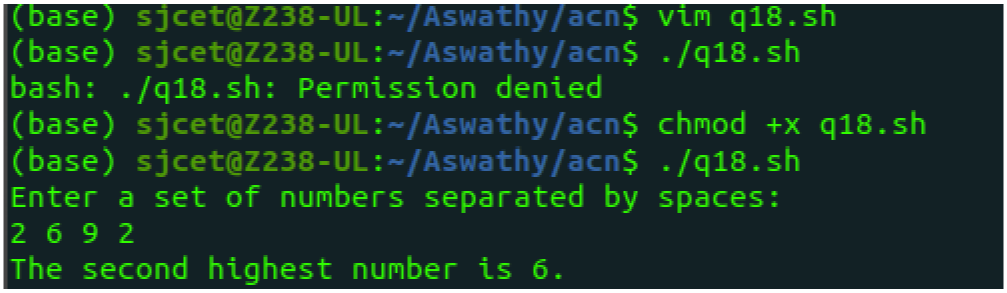
```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q17.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q17.sh
bash: ./q17.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q17.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q17.sh
The sum of all numbers between 50 and 100, which are divisible by 3 and not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

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

<u>Code:</u>

```
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]}."
```

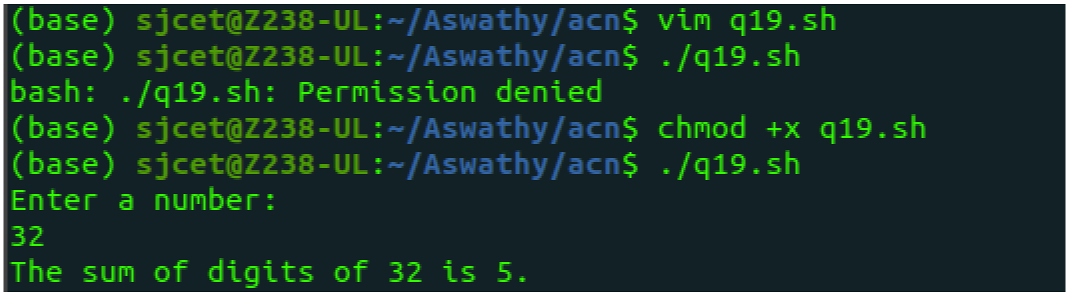


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q18.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q18.sh
bash: ./q18.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q18.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q18.sh
Enter a set of numbers separated by spaces:
2 6 9 2
The second highest number is 6.
```

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

Code:

```
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."
```

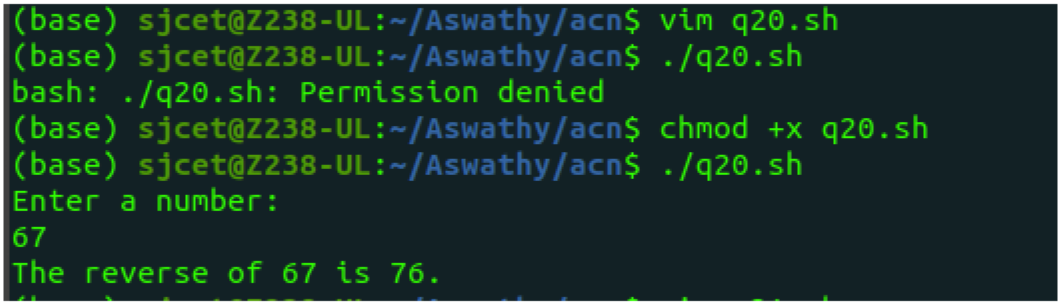


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q19.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q19.sh
bash: ./q19.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q19.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q19.sh
Enter a number:
32
The sum of digits of 32 is 5.
```

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

Code:

```
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."
```

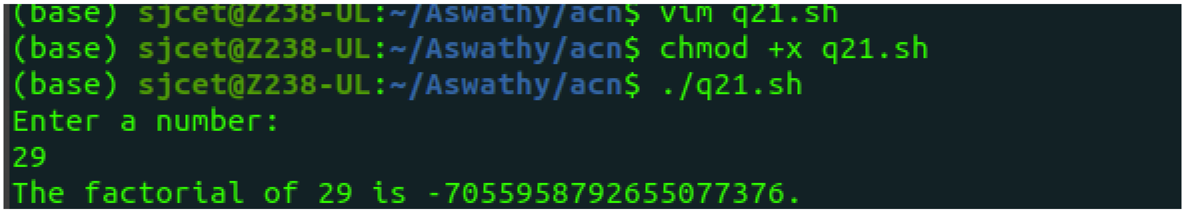


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q20.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q20.sh
bash: ./q20.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q20.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q20.sh
Enter a number:
67
The reverse of 67 is 76.
```

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

Code:

```
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."
```



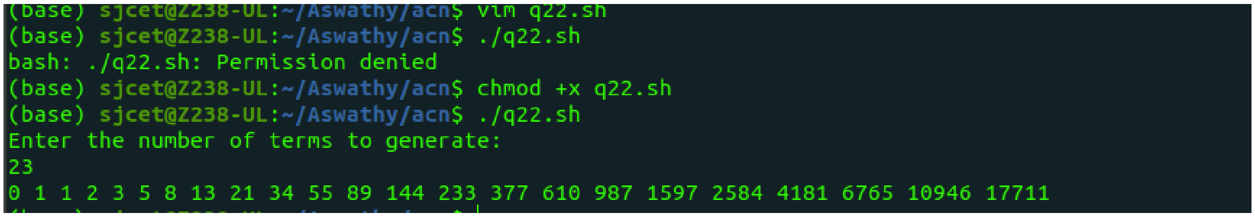
```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q21.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q21.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q21.sh
Enter a number:
29
The factorial of 29 is -7055958792655077376.
```

22. Write a Shell program to generate Fibonacci series.

Code:

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


a=0
b=1
echo -n "$a $b"
for (( i=3; i<=$num; i++ ))
do

   c=$((a + b))
 echo -n " $c"
a=$b
 b=$c
```

```
Done
echo
```

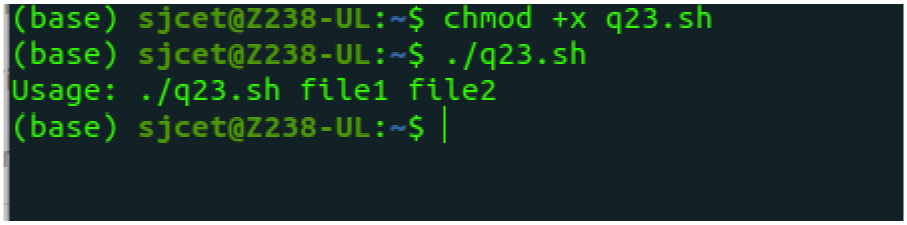


```
(base) sjcet@Z238-UL:~/Aswathy/acn$ vim q22.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q22.sh
bash: ./q22.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn$ chmod +x q22.sh
(base) sjcet@Z238-UL:~/Aswathy/acn$ ./q22.sh
Enter the number of terms to generate:
23
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711
```

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.

Code:

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

if cmp -s "$1" "$2"; then
  rm "$2"
  echo "File '$2' deleted because its contents were identical to '$1'"
else
  echo "File '$2' was not deleted because its contents differed from '$1'"
fi
```

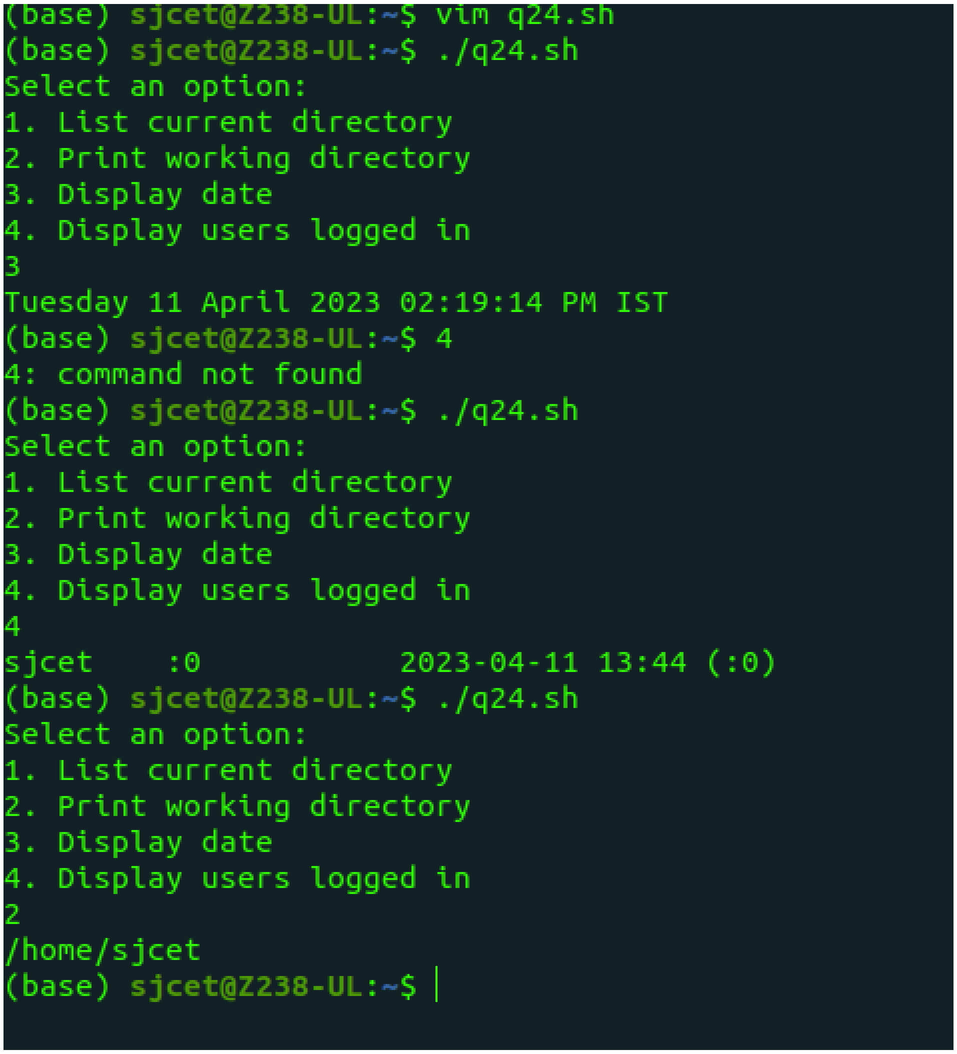


```
(base) sjcet@Z238-UL:~$ chmod +x q23.sh
(base) sjcet@Z238-UL:~$ ./q23.sh
Usage: ./q23.sh file1 file2
(base) sjcet@Z238-UL:~$ |
```

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

Code:

```
while true; do
    clear
    echo "======================="
    echo "      MAIN MENU        "
    echo "======================="
    echo "1. List current directory"
    echo "2. Print working directory"
    echo "3. Display date"
    echo "4. Display users logged in"
    echo "5. Exit"
    echo -n "Enter your choice: "
    read choice

    case $choice in
        1)
            ls -la
            echo "Press enter to continue"
            read
            ;;
        2)
            pwd
            echo "Press enter to continue"
            read
            ;;
        3)
            date
            echo "Press enter to continue"
            read
            ;;
        4)
            who
            echo "Press enter to continue"
            read
            ;;
        5)
            echo "Exiting..."
            exit 0
            ;;
        *)
            echo "Invalid choice. Press enter to continue"
            read
            ;;
    esac
done
```

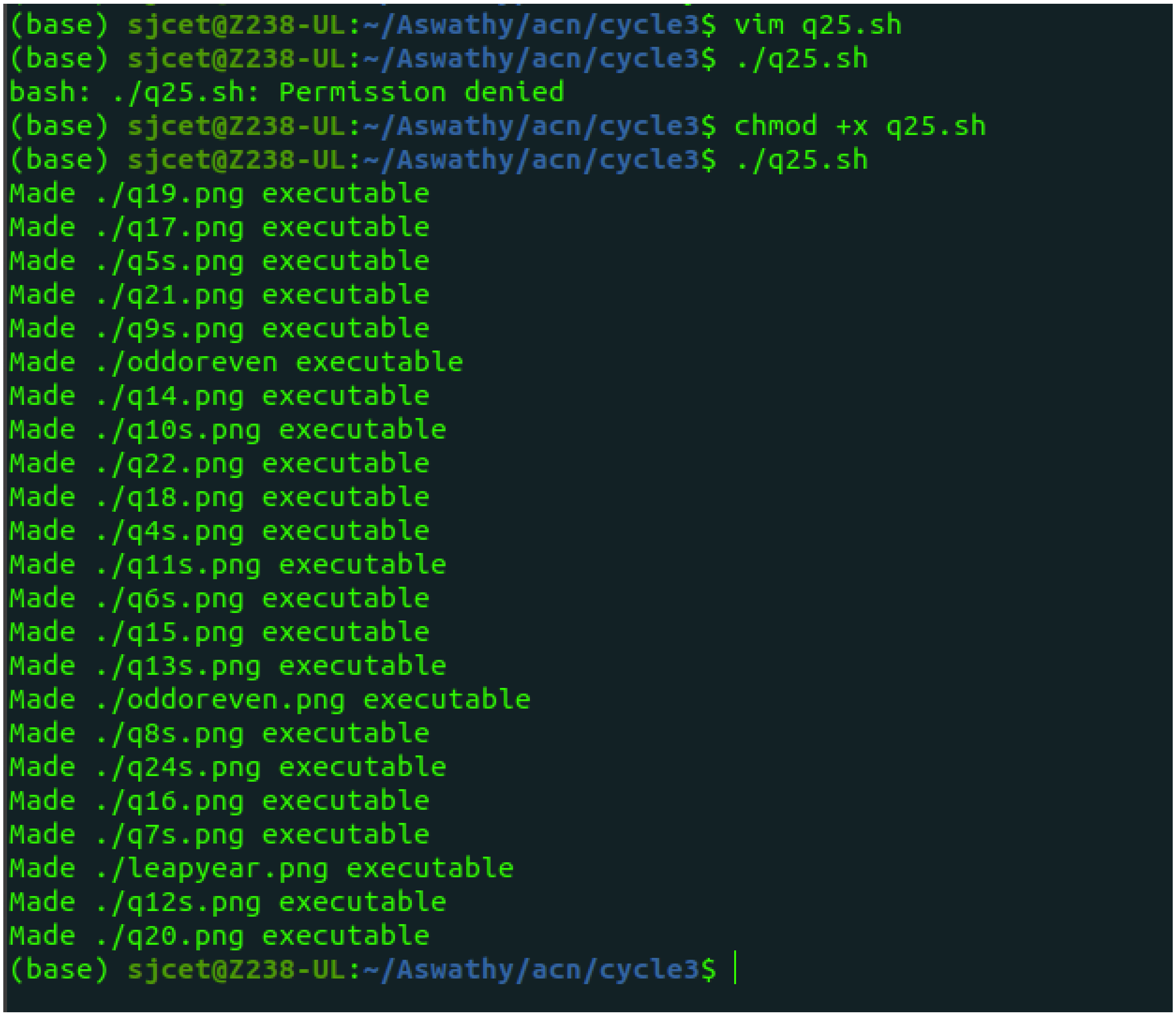


```
(base) sjcet@Z238-UL:~$ vim q24.sh
(base) sjcet@Z238-UL:~$ ./q24.sh
Select an option:
1. List current directory
2. Print working directory
3. Display date
4. Display users logged in
3
Tuesday 11 April 2023 02:19:14 PM IST
(base) sjcet@Z238-UL:~$ 4
4: command not found
(base) sjcet@Z238-UL:~$ ./q24.sh
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 13:44 (:0)
(base) sjcet@Z238-UL:~$ ./q24.sh
Select an option:
1. List current directory
2. Print working directory
3. Display date
4. Display users logged in
2
/home/sjcet
(base) sjcet@Z238-UL:~$ 
```

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.

Code:

```
for file in *; do

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

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

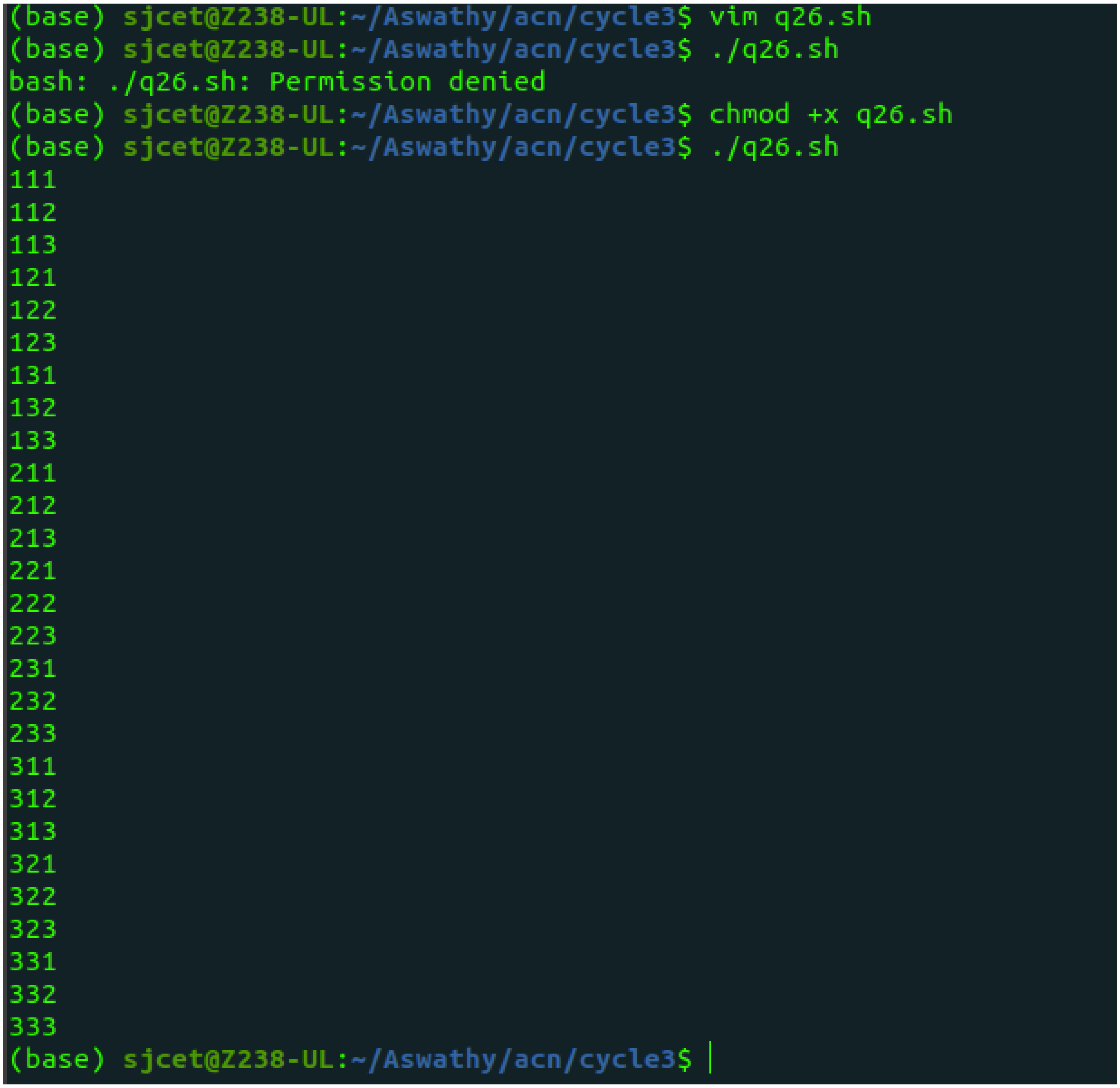


```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ vim q25.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q25.sh
bash: ./q25.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ chmod +x q25.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q25.sh
Made ./q19.png executable
Made ./q17.png executable
Made ./q5s.png executable
Made ./q21.png executable
Made ./q9s.png executable
Made ./oddoreven executable
Made ./q14.png executable
Made ./q10s.png executable
Made ./q22.png executable
Made ./q18.png executable
Made ./q4s.png executable
Made ./q11s.png executable
Made ./q6s.png executable
Made ./q15.png executable
Made ./q13s.png executable
Made ./oddoreven.png executable
Made ./q8s.png executable
Made ./q24s.png executable
Made ./q16.png executable
Made ./q7s.png executable
Made ./leapyear.png executable
Made ./q12s.png executable
Made ./q20.png executable
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ 
```

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

Code:

```
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
```



```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ vim q26.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q26.sh
bash: ./q26.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ chmod +x q26.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./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
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ 
```

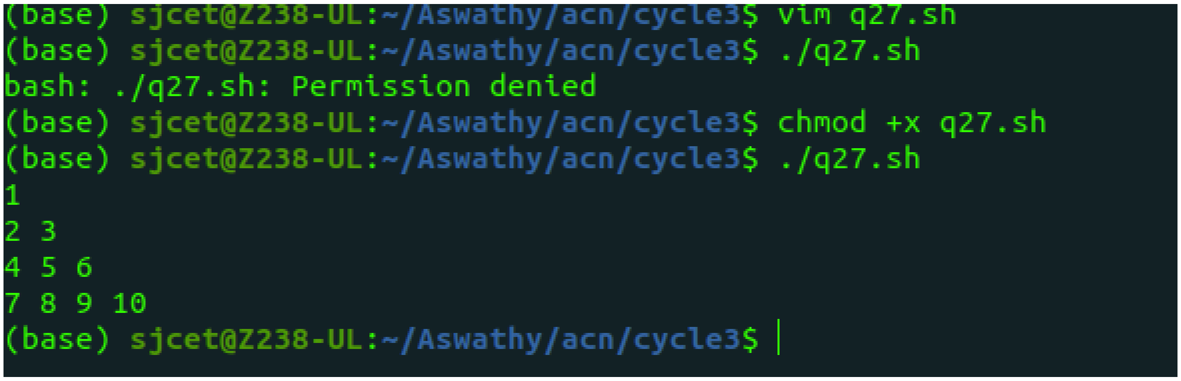
27. Write a Shell program to create the number series.

1
2 3
4 5 6
7 8 9 10

Code:

```
rows=4
current=1
```

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

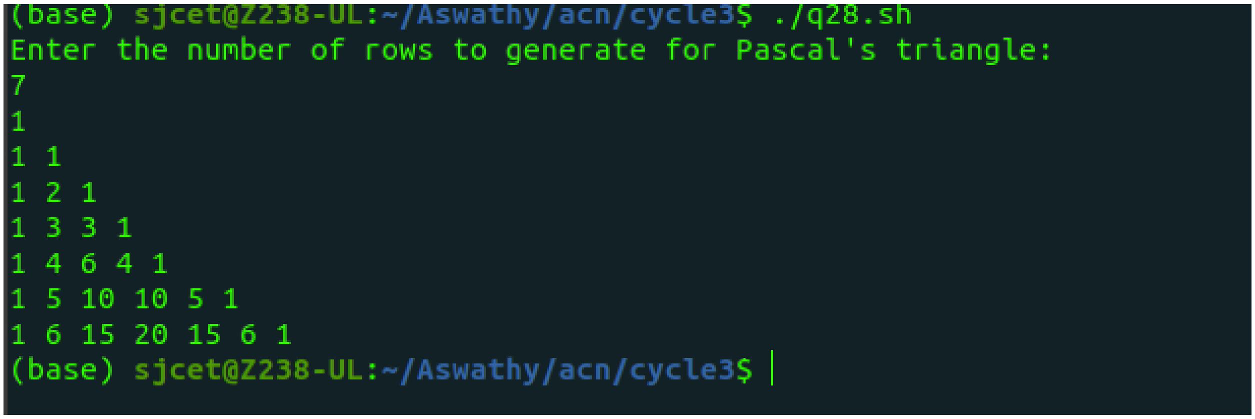


```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ vim q27.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q27.sh
bash: ./q27.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ chmod +x q27.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q27.sh
1
2 3
4 5 6
7 8 9 10
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$
```

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

Code:

```
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


for (( i=0; i<$rows; i++ )); do

  for (( j=0; j<=$i; j++ )); do

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

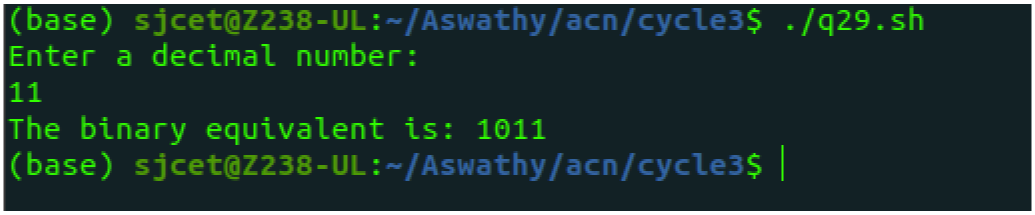


```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q28.sh
Enter the number of rows to generate for Pascal's triangle:
7
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$
```

29. Write a Decimal to Binary Conversion Shell Script.

Code:

```
read -p "Enter decimal number: " decimal
binary=$(echo "obase=2;$decimal" | bc)
echo "Binary equivalent of $decimal is: $binary"
```



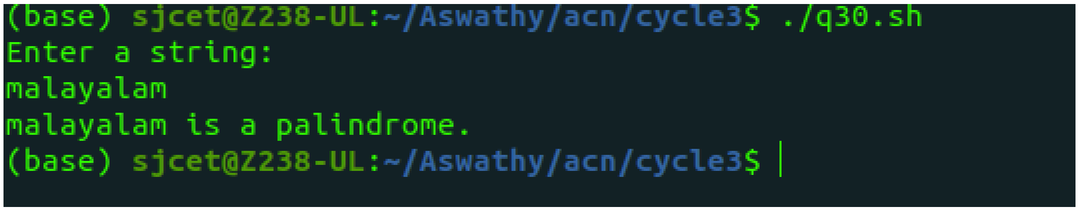
```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q29.sh
Enter a decimal number:
11
The binary equivalent is: 1011
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$
```

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

Code:

```
echo "Enter a string: "
read string


reverse=$(echo $string | rev)


if [ "$string" == "$reverse" ]
then
```

```
    echo "The string is a palindrome"
else
    echo "The string is not a palindrome"
fi
```

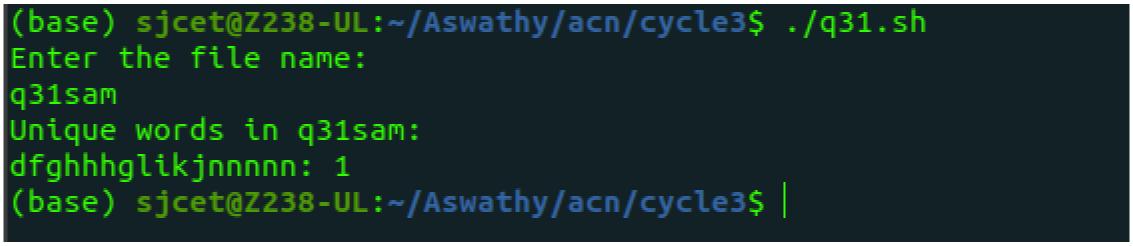


```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q30.sh
Enter a string:
malayalam
malayalam is a palindrome.
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ 
```

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

Code:

```
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
```

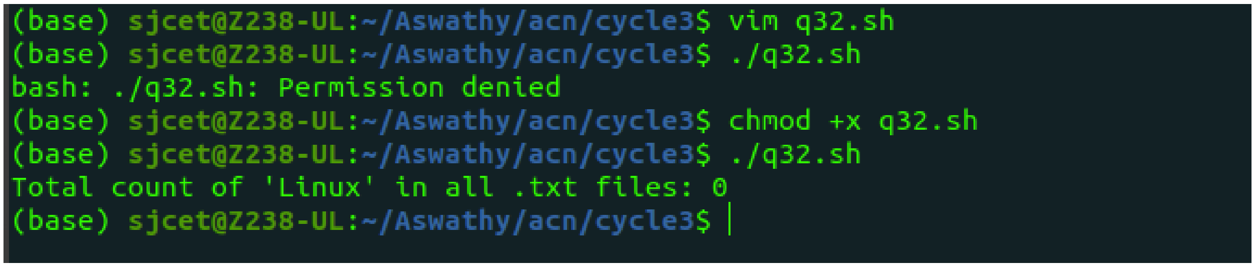


```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q31.sh
Enter the file name:
q31sam
Unique words in q31sam:
dfghhhglikjnnnnn: 1
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ |
```

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.

Code:

```
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"
```



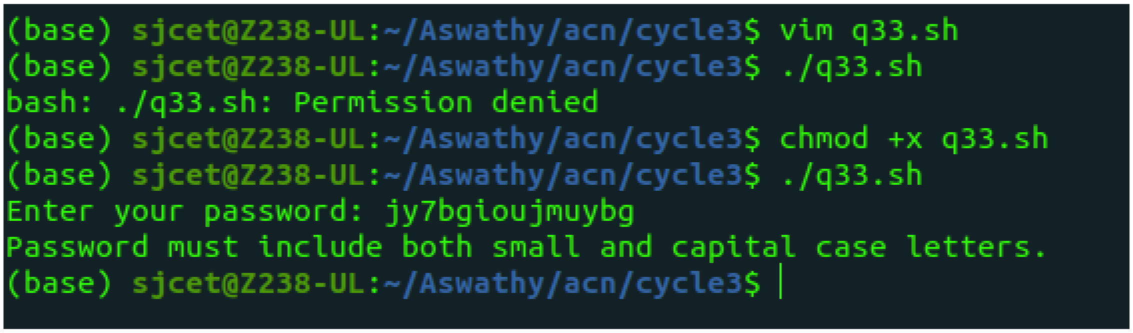
```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ vim q32.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q32.sh
bash: ./q32.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ chmod +x q32.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q32.sh
Total count of 'Linux' in all .txt files: 0
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ |
```

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.

Code:

```
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."
```



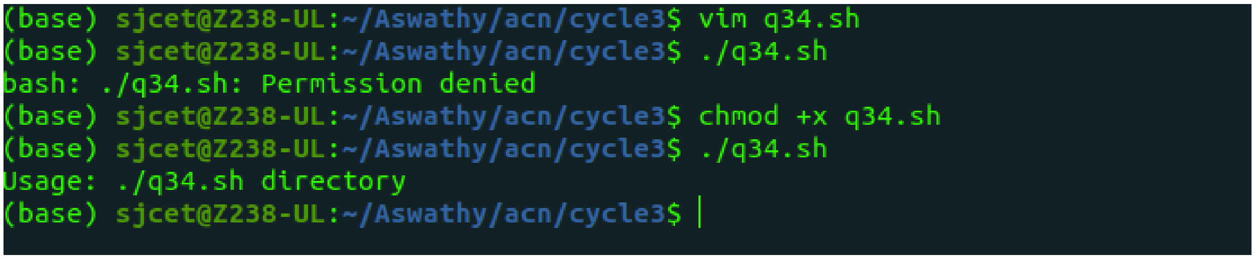
```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ vim q33.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q33.sh
bash: ./q33.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ chmod +x q33.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q33.sh
Enter your password: jy7bgioujmuybg
Password must include both small and capital case letters.
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$
```

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

Code:

```
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"
```

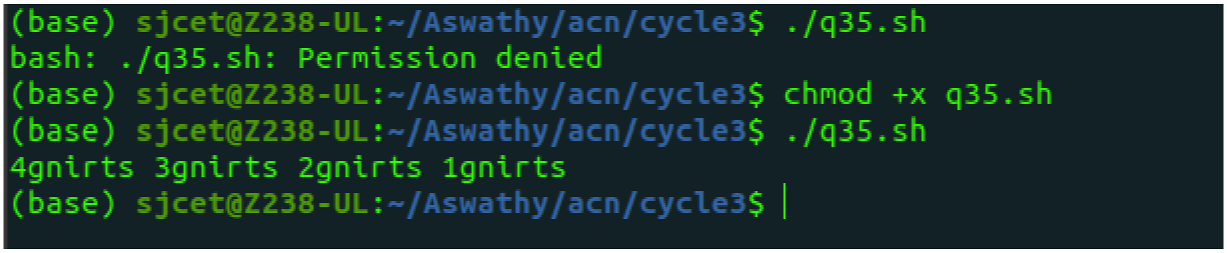


```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ vim q34.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q34.sh
bash: ./q34.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ chmod +x q34.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q34.sh
Usage: ./q34.sh directory
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ 
```

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

<u>Code:</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[@]}"
```



```
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q35.sh
bash: ./q35.sh: Permission denied
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ chmod +x q35.sh
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ ./q35.sh
4gnirts 3gnirts 2gnirts 1gnirts
(base) sjcet@Z238-UL:~/Aswathy/acn/cycle3$ 
```