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

**program:**

echo -n "Enter a number : "

read n

if [ `expr $n % 2` -eq 0 ]

then

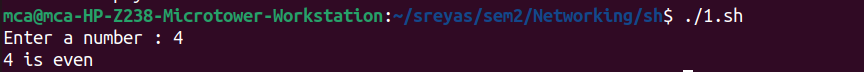
echo "$n is even"

else

echo "$n is odd"

fi

**output:**



2) Write a Shell program to check a leap year.

**program:**

echo -n "Enter year : "

read n

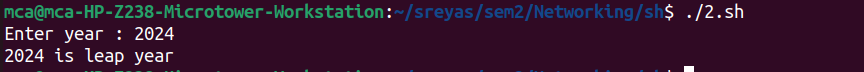
if [ `expr $n % 4` -eq 0 ] && [ `expr $n % 100` -ne 0 ] || [ `expr $n % 400` -eq 0 ]; then

echo "$n is leap year"

else

echo "$n is not a leap year"

**output:**

****

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

**program:**

echo "Enter the radius of the circle:"

read radius

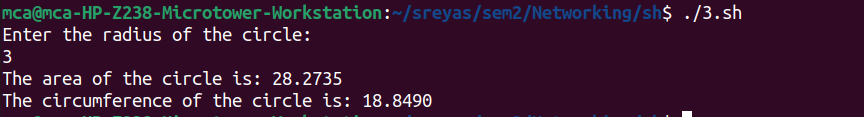
area=$(echo "3.1415 \* ($radius ^ 2)" | bc)

circumference=$(echo "2 \* 3.1415 \* $radius" | bc)

echo "The area of the circle is: $area"

echo "The circumference of the circle is: $circumference"

**output:**

****

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

**program:**

echo "Enter a number:"

read number

reverse=$(echo $number | rev)

if [ $number -eq $reverse ]; then

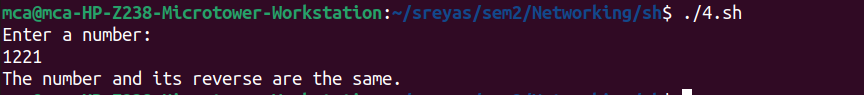
echo "The number and its reverse are the same."

else

echo "The number and its reverse are not the same."

fi

**output:**

****

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

**program:**

echo "Enter a string:"

read string

reverse=$(echo $string | rev)

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

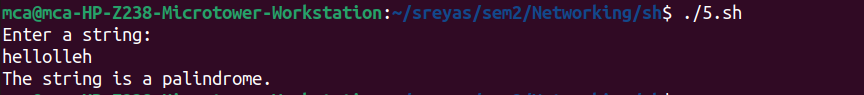
echo "The string is a palindrome."

else

echo "The string is not a palindrome."

fi

**output:**

****

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

**program:**

echo "Enter numbers separated by space:"

read -a numbers

sum\_even=0

sum\_odd=0

for num in "${numbers[@]}"; do

if [ $((num % 2)) -eq 0 ]; then

sum\_even=$((sum\_even + num))

else

sum\_odd=$((sum\_odd + num))

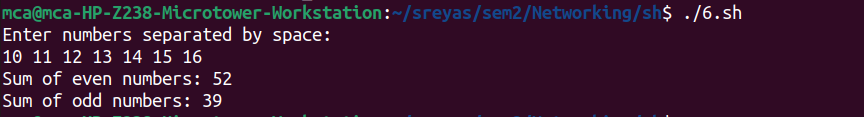
fi

done

echo "Sum of even numbers: $sum\_even"

echo "Sum of odd numbers: $sum\_odd"

**output:**



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

**program:**

echo "Enter the coefficients (a, b, c) of the quadratic equation (ax^2 + bx + c):"

read a b c

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

if [ $discriminant -gt 0 ]; then

root1=$(echo "(-$b + sqrt($discriminant)) / (2 \* $a)" | bc -l)

root2=$(echo "(-$b - sqrt($discriminant)) / (2 \* $a)" | bc -l)

echo "The roots are real and different."

echo "Root 1 = $root1, Root 2 = $root2"

elif [ $discriminant -eq 0 ]; then

root=$(echo "-$b / (2 \* $a)" | bc -l)

echo "The roots are real and equal."

echo "Root 1 = Root 2 = $root"

else

real\_part=$(echo "-$b / (2 \* $a)" | bc -l)

imaginary\_part=$(echo "sqrt($((-1 \* discriminant))) / (2 \* $a)" | bc -l)

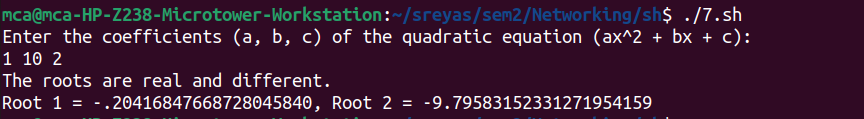
echo "The roots are complex and different."

echo "Root 1 = $real\_part + $imaginary\_part i"

echo "Root 2 = $real\_part - $imaginary\_part i"

fi

**output:**

****

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

**program:**

echo "Enter a number:"

read number

length=${#number}

sum=0

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

digit=${number:i:1}

sum=$((sum + digit \*\* length))

done

if [ $sum -eq $number ]; then

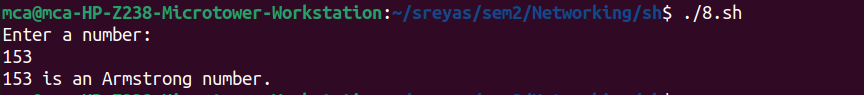
echo "$number is an Armstrong number."

else

echo "$number is not an Armstrong number."

fi

**output:**

****

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

**program:**

echo "Enter a number:"

read number

is\_prime=true

if [ $number -lt 2 ]; then

is\_prime=false

fi

for ((i=2; i<=number/2; i++)); do

if [ $((number % i)) -eq 0 ]; then

is\_prime=false

break

fi

done

if $is\_prime; then

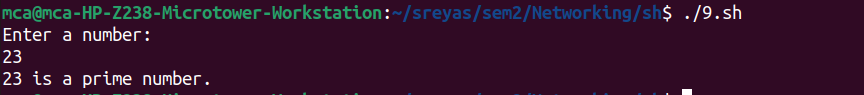
echo "$number is a prime number."

else

echo "$number is not a prime number."

fi

**output:**

****

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

**program:**

echo "Prime numbers between 1 and 50 are:"

for ((i=2; i<=50; i++)); do

is\_prime=true

for ((j=2; j<=i/2; j++)); do

if [ $((i % j)) -eq 0 ]; then

is\_prime=false

break

fi

done

if $is\_prime; then

echo $i

fi

done

**output:**

****

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

**program:**

echo "Enter a number:"

read num

sum=0

while [ $num -gt 0 ]; do

digit=$(( $num % 10 ))

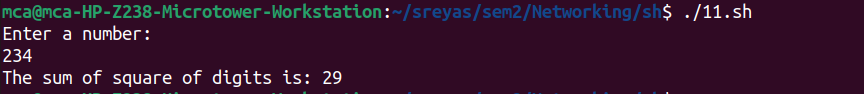
sum=$(( $sum + $digit \* $digit ))

num=$(( $num / 10 ))

done

echo "The sum of square of digits is: $sum"

**output:**

****

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

**program:**

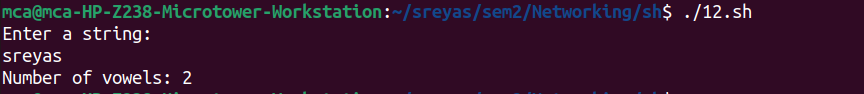
echo "Enter a string:"

read str

count=$(echo $str | grep -o -i "[aeiou]" | wc -l)

echo "Number of vowels: $count"

**output:**

****

13) Write a Shell program to display student grades.

**program:**

calculate\_grade() {

if [ $1 -ge 90 ]; then

grade="A"

elif [ $1 -ge 80 ]; then

grade="B"

elif [ $1 -ge 70 ]; then

grade="C"

elif [ $1 -ge 60 ]; then

grade="D"

else

grade="F"

fi

echo $grade

}

echo "Enter student name:"

read name

echo "Enter student's mark:"

read mark

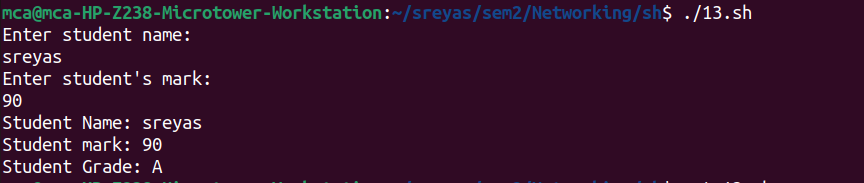
grade=$(calculate\_grade $mark)

echo "Student Name: $name"

echo "Student mark: $mark"

echo "Student Grade: $grade"

**output:**



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

**program:**

echo "Enter numbers separated by space:"

read -a numbers

largest=${numbers[0]}

smallest=${numbers[0]}

for num in "${numbers[@]}"; do

if [ $num -gt $largest ]; then

largest=$num

fi

if [ $num -lt $smallest ]; then

smallest=$num

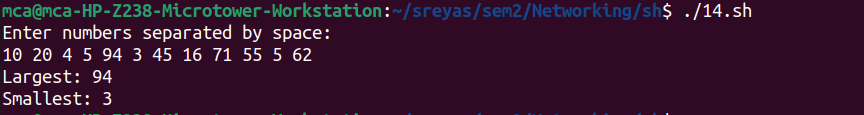
fi

done

echo "Largest: $largest"

echo "Smallest: $smallest"

**output:**

****

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

**program:**

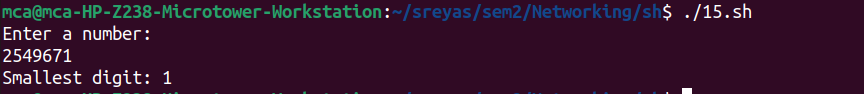
echo "Enter a number:"

read num

smallest=$(echo $num | grep -o "[0-9]" | sort | head -n1)

echo "Smallest digit: $smallest"

**output:**



16) 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.

**program:**

sum=0

for ((i=50; i<=100; i++)); do

if [ $((i % 3)) -eq 0 ] && [ $((i % 5)) -ne 0 ]; then

sum=$((sum + i))

fi

done

echo "Sum of numbers divisible by 3 and not by 5 between 50 and 100: $sum"

**output:**



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

**program:**

echo "Enter numbers separated by space:"

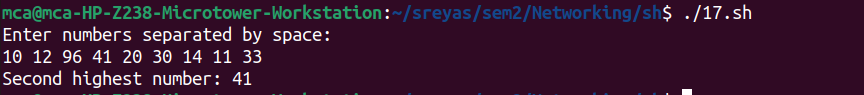
read -a numbers

IFS=$'\n' sorted=($(sort -n <<<"${numbers[\*]}"))

len=${#sorted[@]}

echo "Second highest number: ${sorted[len-2]}"

**output:**

****

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

**program:**

echo "Enter a number:"

read num

sum\_digits() {

local n=$1

local sum=0

while [ $n -gt 0 ]; do

sum=$((sum + n % 10))

n=$((n / 10))

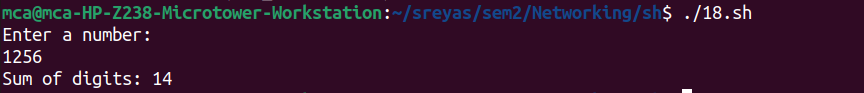
done

echo $sum

}

echo "Sum of digits: $(sum\_digits $num)"

**output:**

****

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

**program:**

echo "Enter a number:"

read num

reverse() {

local n=$1

local rev=0

while [ $n -gt 0 ]; do

remainder=$((n % 10))

rev=$((rev \* 10 + remainder))

n=$((n / 10))

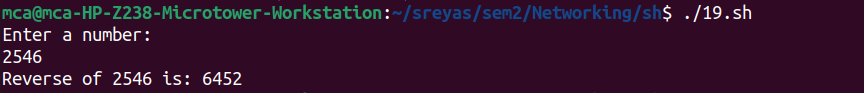
done

echo $rev

}

echo "Reverse of $num is: $(reverse $num)"

**output:**



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

**program:**

echo "Enter a number:"

read num

fact=1

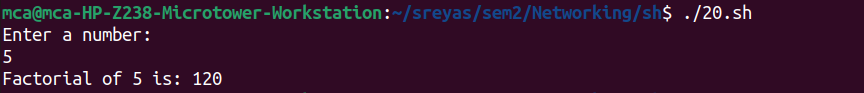
for ((i=1; i<=num; i++)); do

fact=$((fact \* i))

done

echo "Factorial of $num is: $fact"

**output:**



21) Write a Shell program to generate Fibonacci series.

22) 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.

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

24) 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.

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

26) Write a Shell program to create the number series.

27) Write a Shell program to create Pascal’s triangle.

28) Write a Decimal to Binary Conversion Shell Script.

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

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

31) 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.

32) 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.)

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

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