Assignment-3

1. Write a Shell Script to find maximum between two numbers.

#!/bin/bash

# Prompt the user to enter the first number

echo "Enter the first number:"

read num1

# Prompt the user to enter the second number

echo "Enter the second number:"

read num2

# Check if num1 is greater than num2

if [ "$num1" -gt "$num2" ]; then

echo "$num1 is greater than $num2"

elif [ "$num1" -eq "$num2" ]; then

echo "$num1 and $num2 are equal"

else

echo "$num2 is greater than $num1"

fi

1. Write a Shell Script to find maximum between three numbers.

#!/bin/bash

# Prompt the user to enter the three numbers

echo "Enter the first number:"

read num1

echo "Enter the second number:"

read num2

echo "Enter the third number:"

read num3

# Compare the numbers using conditional statements

if [ $num1 -ge $num2 ] && [ $num1 -ge $num3 ]; then

echo "The maximum number is: $num1"

elif [ $num2 -ge $num1 ] && [ $num2 -ge $num3 ]; then

echo "The maximum number is: $num2"

else

echo "The maximum number is: $num3"

fi

1. Write a Shell Script to check whether a number is negative, positive or zero.

#!/bin/bash

# Prompt the user to enter a number

echo "Enter a number:"

read num

# Check if the number is negative

if [ $num -lt 0 ]; then

echo "The number is negative."

# Check if the number is positive

elif [ $num -gt 0 ]; then

echo "The number is positive."

# If the number is not negative or positive, it must be zero

else

echo "The number is zero."

Fi

1. Write a Shell Script to check whether a number is divisible by 5 and 11 or not.

#!/bin/bash

# Prompt the user to enter a number

echo "Enter a number:"

read num

# Check if the number is divisible by both 5 and 11

if [ $((num % 5)) -eq 0 ] && [ $((num % 11)) -eq 0 ]; then

echo "The number $num is divisible by both 5 and 11."

else

echo "The number $num is not divisible by both 5 and 11."

Fi

1. Write a Shell Script to check whether a number is even or odd.

#!/bin/bash

# Prompt the user to enter a number

echo "Enter a number:"

read num

# Check if the number is even

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

echo "The number $num is even."

else

echo "The number $num is odd."

Fi

1. Write a Shell Script to check whether a year is leap year or not.

#!/bin/bash

# Prompt the user to enter a year

echo "Enter a year:"

read year

# Check if the year is a leap year

if [ $((year % 4)) -eq 0 ] && [ $((year % 100)) -ne 0 ] || [ $((year % 400)) -eq 0 ]; then

echo "The year $year is a leap year."

else

echo "The year $year is not a leap year."

Fi

1. Write shell script to check eligibility of candidate for voter id card

#!/bin/bash

# Prompt the user to enter their age

echo "Enter your age:"

read age

# Check if the age is eligible for a voter ID card

if [ $age -ge 18 ]; then

echo "Congratulations! You are eligible for a voter ID card."

else

echo "Sorry, you are not eligible for a voter ID card yet. You must be at least 18 years old."

Fi

1. Shell Script to display the first 10 natural numbers.

Expected Output :

1 2 3 4 5 6 7 8 9 10

#!/bin/bash

# Loop from 1 to 10 and display each number

for (( i=1; i<=10; i++ )); do

echo -n "$i " # "-n" option prevents the echo command from adding a newline character

done

echo # Add a newline character after displaying all numbers

1. Shell Script to compute the sum of the first 10 natural numbers.

Expected Output :

The first 10 natural number is :

1 2 3 4 5 6 7 8 9 10

The Sum is : 55

1. Shell Script to display n terms of natural numbers and their sum.

Test Data : 7

Expected Output :

The first 7 natural number is :

1 2 3 4 5 6 7

The Sum of Natural Number upto 7 terms : 28

#!/bin/bash

# Initialize variables

sum=0

numbers=""

# Loop from 1 to 10 to compute the sum and construct the string of numbers

for (( i=1; i<=10; i++ )); do

sum=$((sum + i))

numbers="$numbers $i"

done

# Print the first 10 natural numbers

echo "The first 10 natural numbers are :$numbers"

# Print the sum of the first 10 natural numbers

echo "The Sum is : $sum"

1. Shell Script to read 10 numbers from the keyboard and find their sum and average.

Test Data :

Input the 10 numbers :

Number-1 :2

...

Number-10 :2

Expected Output :

The sum of 10 no is : 55

The Average is : 5.500000

1. Shell Script to display the cube of the number up to an integer.

Test Data :

Input number of terms : 5

Expected Output :

Number is : 1 and cube of the 1 is :1

Number is : 2 and cube of the 2 is :8

Number is : 3 and cube of the 3 is :27

Number is : 4 and cube of the 4 is :64

Number is : 5 and cube of the 5 is :125

#!/bin/bash

# Initialize variables

sum=0

average=0

# Prompt the user to input 10 numbers

echo "Input the 10 numbers:"

# Loop to read each number and calculate the sum

for (( i=1; i<=10; i++ )); do

echo -n "Number-$i: "

read num

sum=$((sum + num))

done

# Calculate the average

average=$(echo "scale=6; $sum / 10" | bc) # Using bc to handle decimal arithmetic

# Print the sum and average

echo "The sum of 10 numbers is: $sum"

echo "The Average is: $average"

1. Shell Script to display the multiplication table for a given integer.

Test Data :

Input the number (Table to be calculated) : 15

Expected Output :

15 X 1 = 15

...

...

15 X 10 = 150

#!/bin/bash

# Prompt the user to input the number for which multiplication table is to be calculated

echo "Input the number (Table to be calculated):"

read number

# Display the multiplication table for the given number

echo "Multiplication table for $number:"

# Loop from 1 to 10 to calculate and display each multiplication

for (( i=1; i<=10; i++ )); do

echo "$number X $i = $((number \* i))"

done

1. Shell Script to display the multiplier table vertically from 1 to n.

Test Data :

Input upto the table number starting from 1 : 8

Expected Output :

Multiplication table from 1 to 8

1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8

...

1x10 = 10, 2x10 = 20, 3x10 = 30, 4x10 = 40, 5x10 = 50, 6x10 = 60, 7x10 = 70, 8x10 = 80

#!/bin/bash

# Prompt the user to input the maximum table number

echo "Input up to the table number starting from 1:"

read max\_number

# Display the multiplication table from 1 to the maximum number

echo "Multiplication table from 1 to $max\_number:"

# Loop through each multiplier from 1 to 10

for (( i=1; i<=10; i++ )); do

# Loop through each number from 1 to the maximum table number

for (( j=1; j<=$max\_number; j++ )); do

# Calculate and display the multiplication for the current multiplier and number

echo -n "$j x $i = $((j \* i)), "

done

echo # Move to the next line after displaying all multiplications for the current multiplier

done

1. Shell Script to display the n terms of odd natural numbers and their sum.

Test Data

Input number of terms : 10

Expected Output :

The odd numbers are :1 3 5 7 9 11 13 15 17 19

The Sum of odd Natural Number upto 10 terms : 100

#!/bin/bash

# Prompt the user to input the number of terms

echo "Input number of terms:"

read n

# Initialize variables

sum=0

odd\_numbers=""

# Loop through odd natural numbers and calculate sum

for (( i=1; i<=2\*n; i+=2 )); do

odd\_numbers="$odd\_numbers $i" # Append current odd number to the list

sum=$((sum + i)) # Add current odd number to the sum

done

# Display the odd numbers

echo "The odd numbers are:$odd\_numbers"

# Display the sum of odd natural numbers

echo "The Sum of odd Natural Number upto $n terms: $sum"