KARTIK SHARMA :-  (BT23ECE031)

Q1]

# Take user Input

echo "Enter Two numbers : "

read a

read b

# Input type of operation

echo "Enter Choice :"

echo "1. Addition"

echo "2. Subtraction"

echo "3. Multiplication"

echo "4. Division"

read ch

# Switch Case to perform

# calculator operations

case $ch in

1)res=`echo $a + $b | bc`

;;

2)res=`echo $a - $b | bc`

;;

3)res=`echo $a \\* $b | bc`

;;

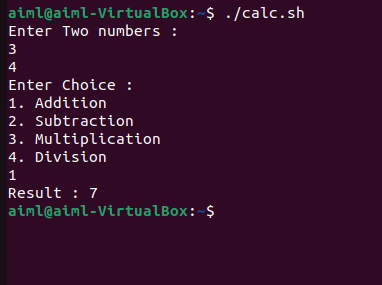
4)res=`echo "scale=2; $a / $b" | bc`

;;

esac

echo "Result : $res"

Execution:-



Q2 ]

# Prompt user for input

read -p "Enter numbers separated by spaces: " -a arr

# Initialize sum variable

sum=0

# Calculate the sum of the array elements

for i in "${arr[@]}"

do

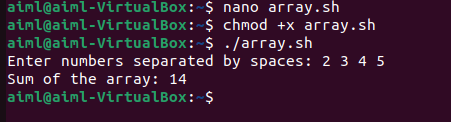
    sum=$((sum + i))

done

# Output the sum

echo "Sum of the array: $sum"

Execution:-



Q3]

# Prompt user for input

read -p "Enter numbers (as strings) separated by spaces: " -a arr

sum=0

# Calculate the sum of the array elements

for i in "${arr[@]}"

do

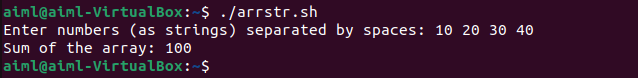
    # Convert string to integer and add to sum

    sum=$((sum + i))

done

echo "Sum of the array: $sum"

Execution:-



Q4]

# Function to count lines in a file

count\_lines() {

    if [[ -f $1 ]]; then  # Check if the file exists

        line\_count=$(wc -l < "$1")  # Count the number of lines

        echo "Number of lines in '$1': $line\_count"

    else

        echo "Error: File '$1' does not exist."

    fi

}

# Check if a file argument is provided

if [[ $# -ne 1 ]]; then

    echo "Usage: $0 <filename>"

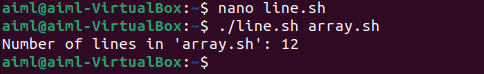
    exit 1

fi

# Call the function with the provided argument

count\_lines "$1"

Execution:-



Q5]

# Function to count lines in a list of files

count\_lines() {

  total\_lines=0

  for file in "$@"; do

    if [ -f "$file" ]; then

      lines=$(wc -l < "$file")

      total\_lines=$((total\_lines + lines))

    else

      echo "File '$file' not found!"

    fi

  done

return $total\_lines  # Return the total line count

}

if [ $# -eq 0 ]; then

  echo "Please provide file names as arguments."

  exit 1

fi

# Call the function with the provided file names

count\_lines "$@"

line\_count=$?

echo "Total number of lines in provided files: $line\_count"

Execution:-



Q6 ]

# Loop through files and directories in the current directory

for item in \*; do

  echo "$item"

done

Execution:-

