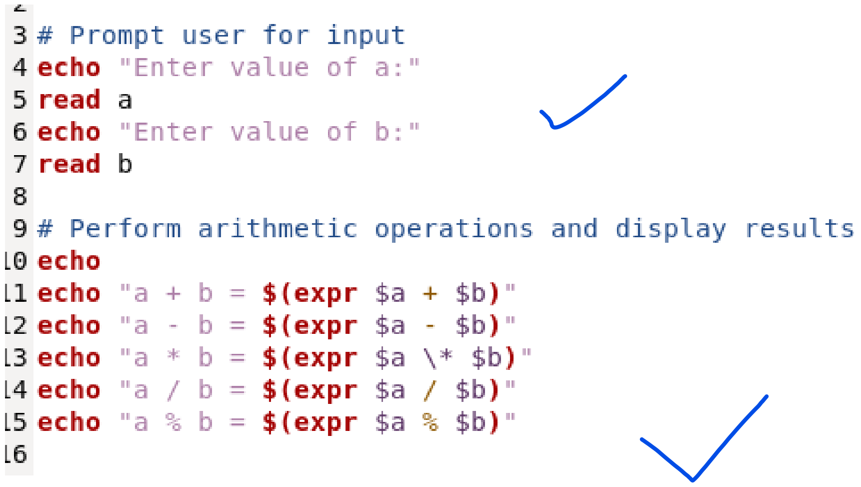
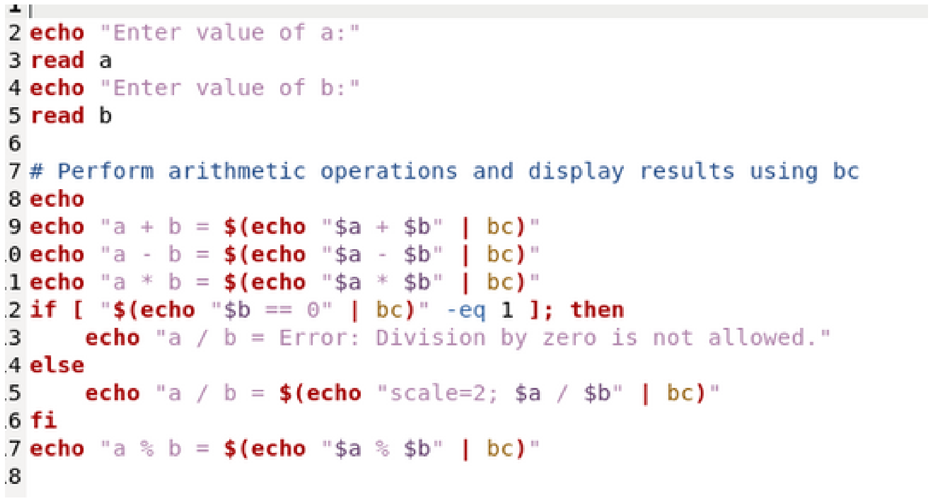


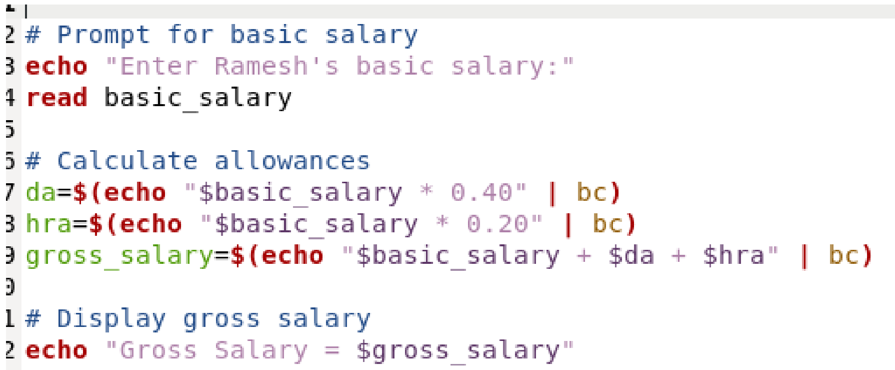
```
# Prompt user for input
echo "Enter value of a:"
read a
echo "Enter value of b:"
read b

# Perform arithmetic operations and display results
echo
echo "a + b = $(expr $a + $b)"
echo "a - b = $(expr $a - $b)"
echo "a * b = $(expr $a \* $b)"
echo "a / b = $(expr $a / $b)"
echo "a % b = $(expr $a % $b)"

```

✓ ✓



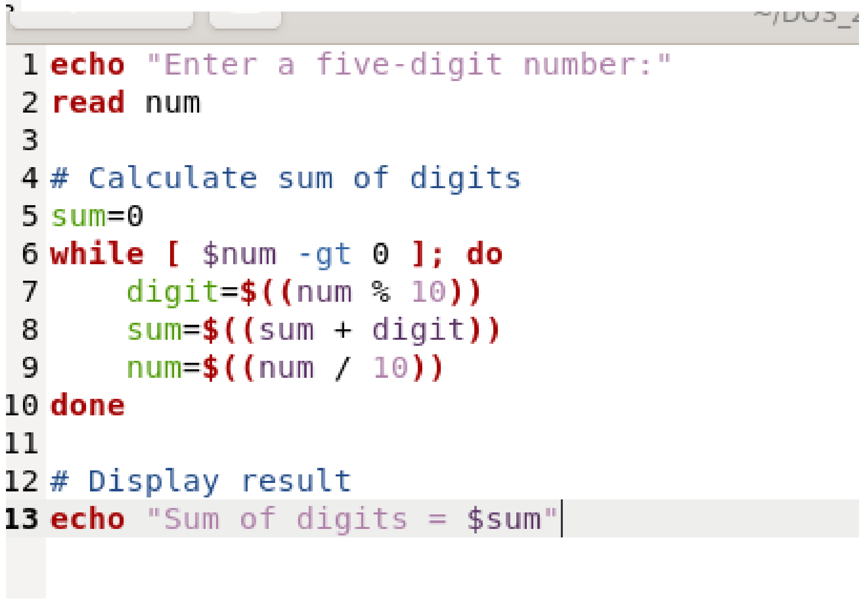
```

echo "Enter value of a:"
read a
echo "Enter value of b:"
read b

# Perform arithmetic operations and display results using bc
echo
echo "a + b = $(echo "$a + $b" | bc)"
echo "a - b = $(echo "$a - $b" | bc)"
echo "a * b = $(echo "$a * $b" | bc)"
if [ "$(echo "$b == 0" | bc)" -eq 1 ]; then
    echo "a / b = Error: Division by zero is not allowed."
else
    echo "a / b = $(echo "scale=2; $a / $b" | bc)"
fi
echo "a % b = $(echo "$a % $b" | bc)"

```

✓



```

# Prompt for basic salary
echo "Enter Ramesh's basic salary:"
read basic_salary

# Calculate allowances
da=$(echo "$basic_salary * 0.40" | bc)
hra=$(echo "$basic_salary * 0.20" | bc)
gross_salary=$(echo "$basic_salary + $da + $hra" | bc)

# Display gross salary
echo "Gross Salary = $gross_salary"
```

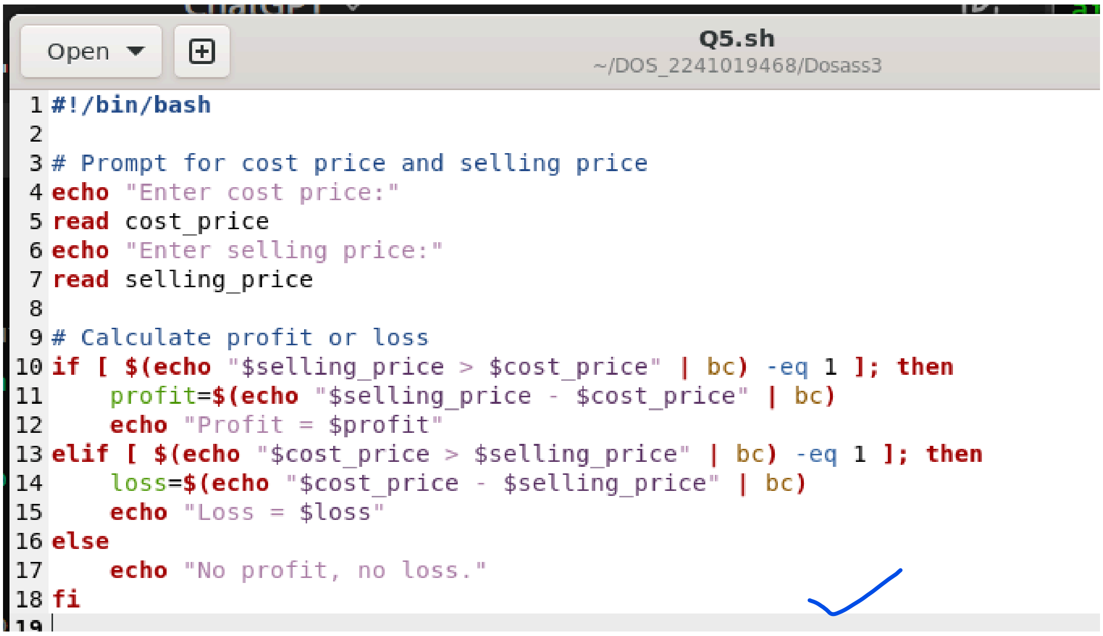
✓



```
echo "Enter a five-digit number:"
read num

# Calculate sum of digits
sum=0
while [ $num -gt 0 ]; do
    digit=$((num % 10))
    sum=$((sum + digit))
    num=$((num / 10))
done

# Display result
echo "Sum of digits = $sum"
```

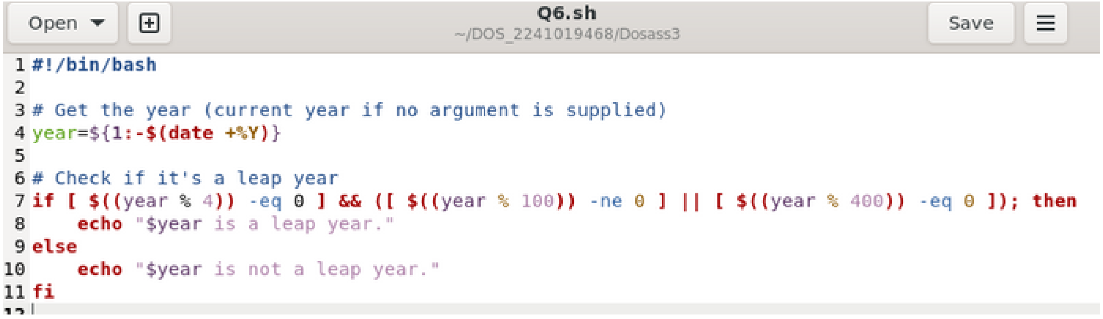
✓

✓



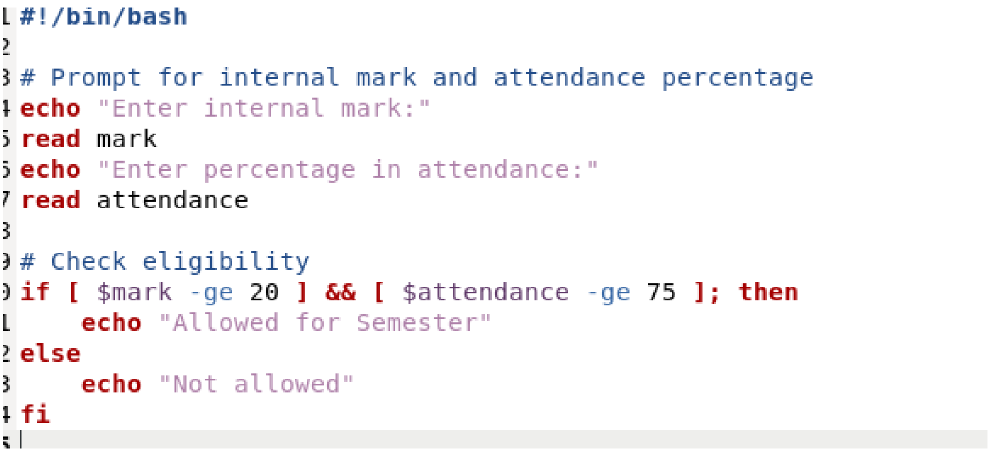
```bash
#!/bin/bash

# Prompt for cost price and selling price
echo "Enter cost price:"
read cost_price
echo "Enter selling price:"
read selling_price

# Calculate profit or loss
if [ $(echo "$selling_price > $cost_price" | bc) -eq 1 ]; then
    profit=$(echo "$selling_price - $cost_price" | bc)
    echo "Profit = $profit"
elif [ $(echo "$cost_price > $selling_price" | bc) -eq 1 ]; then
    loss=$(echo "$cost_price - $selling_price" | bc)
    echo "Loss = $loss"
else
    echo "No profit, no loss."
fi
```

✓



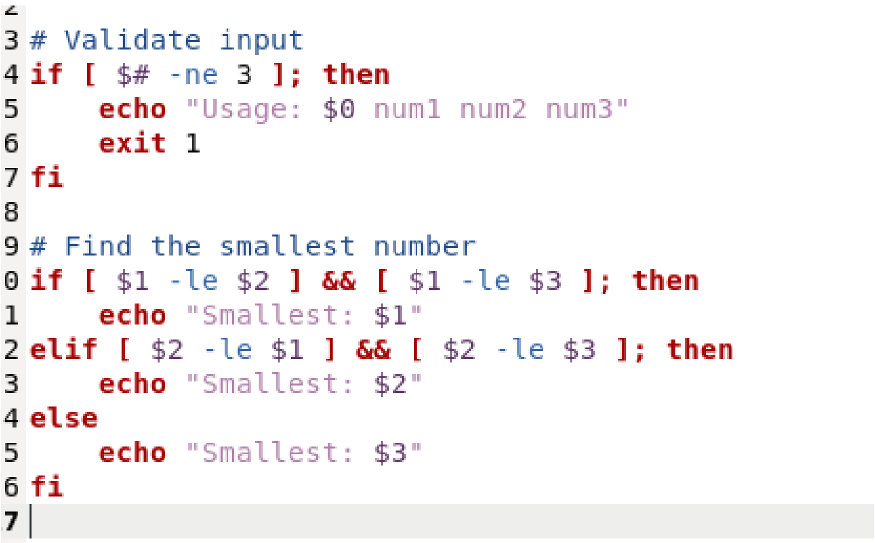
```bash
#!/bin/bash

# Get the year (current year if no argument is supplied)
year=${1:-$(date +%Y)}

# Check if it's a leap year
if [ $((year % 4)) -eq 0 ] && ([ $((year % 100)) -ne 0 ] || [ $((year % 400)) -eq 0 ]); then
    echo "$year is a leap year."
else
    echo "$year is not a leap year."
fi
```

7



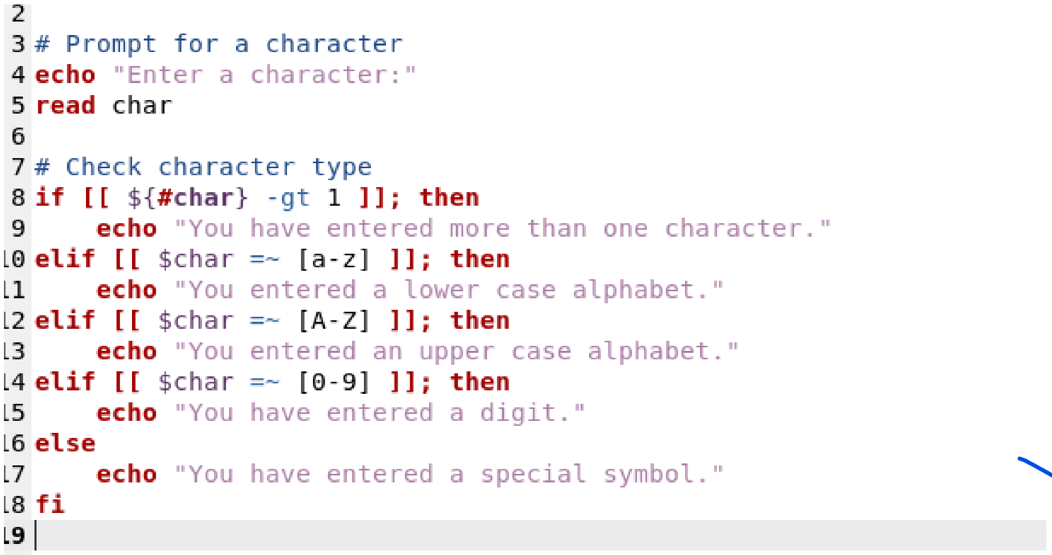
```bash
#!/bin/bash

# Prompt for internal mark and attendance percentage
echo "Enter internal mark:"
read mark
echo "Enter percentage in attendance:"
read attendance

# Check eligibility
if [ $mark -ge 20 ] && [ $attendance -ge 75 ]; then
    echo "Allowed for Semester"
else
    echo "Not allowed"
fi
```

8



```bash
# Validate input
if [ $# -ne 3 ]; then
    echo "Usage: $0 num1 num2 num3"
    exit 1
fi

# Find the smallest number
if [ $1 -le $2 ] && [ $1 -le $3 ]; then
    echo "Smallest: $1"
elif [ $2 -le $1 ] && [ $2 -le $3 ]; then
    echo "Smallest: $2"
else
    echo "Smallest: $3"
fi
```

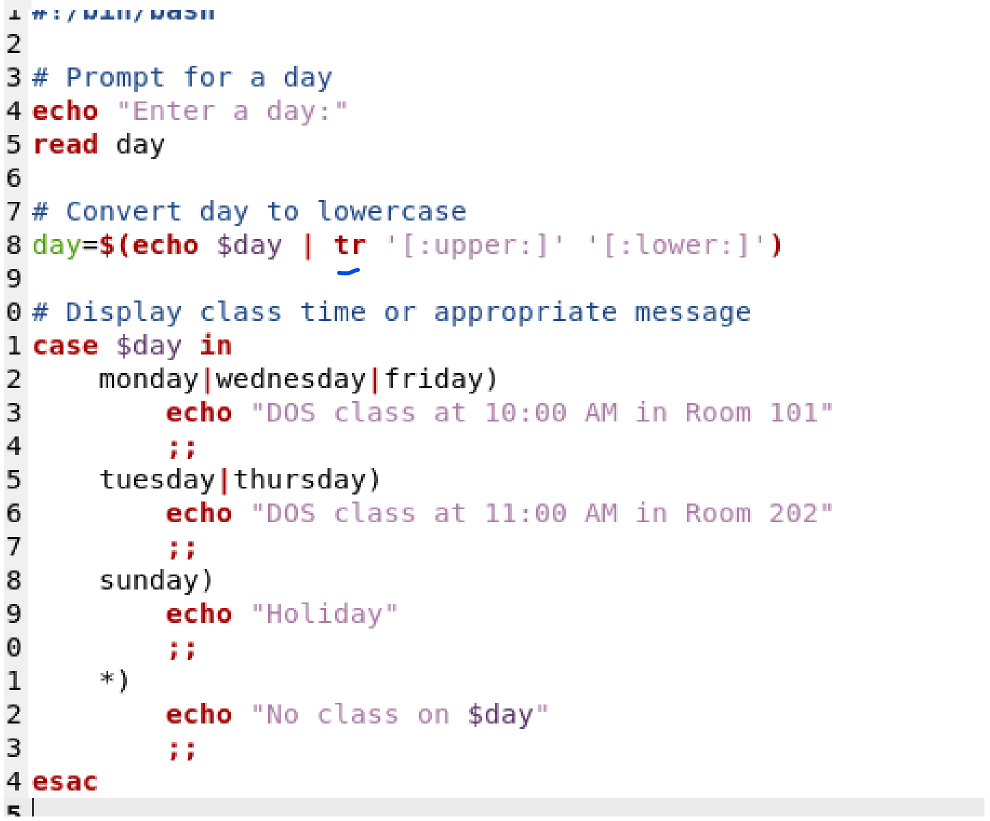
✓



```
# Prompt for a character
echo "Enter a character:"
read char

# Check character type
if [[ ${#char} -gt 1 ]]; then
    echo "You have entered more than one character."
elif [[ $char =~ [a-z] ]]; then
    echo "You entered a lower case alphabet."
elif [[ $char =~ [A-Z] ]]; then
    echo "You entered an upper case alphabet."
elif [[ $char =~ [0-9] ]]; then
    echo "You have entered a digit."
else
    echo "You have entered a special symbol."
fi
```

9

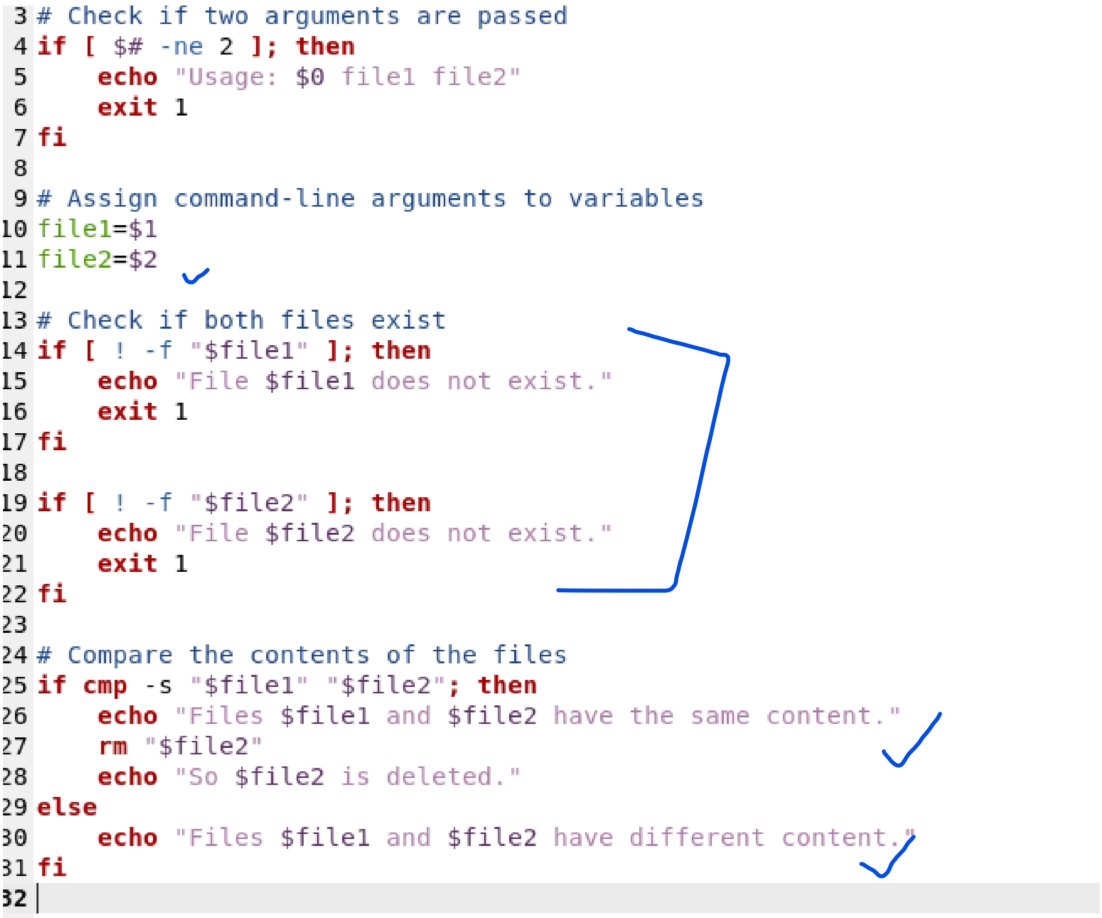
✓



```
#!/bin/bash

# Prompt for a day
echo "Enter a day:"
read day

# Convert day to lowercase
day=$(echo $day | tr '[:upper:]' '[:lower:]')

# Display class time or appropriate message
case $day in
    monday|wednesday|friday)
        echo "DOS class at 10:00 AM in Room 101"
        ;;
    tuesday|thursday)
        echo "DOS class at 11:00 AM in Room 202"
        ;;
    sunday)
        echo "Holiday"
        ;;
    *)
        echo "No class on $day"
        ;;
esac
```

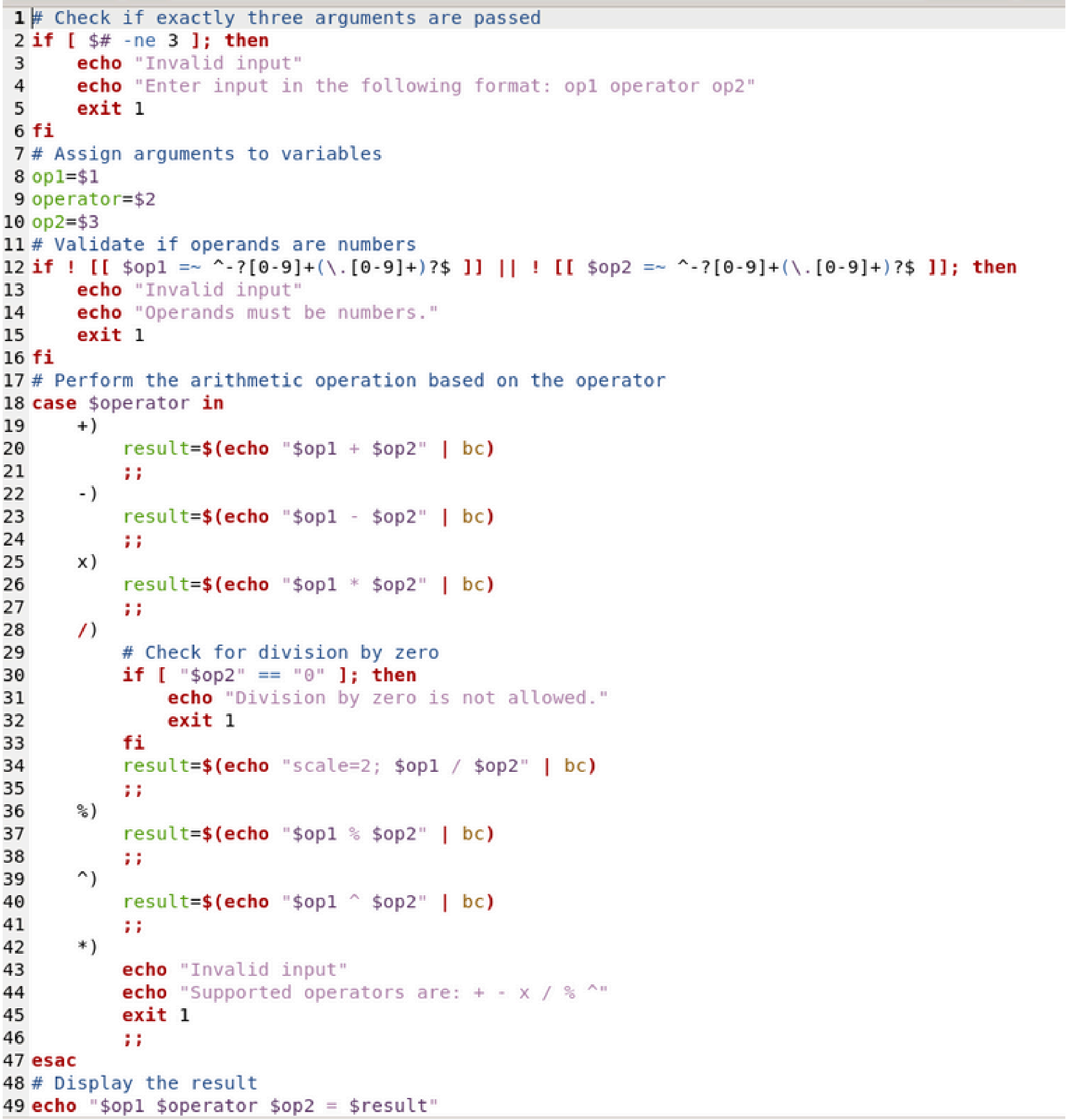
10

✓



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

# Assign command-line arguments to variables
file1=$1
file2=$2

# Check if both files exist
if [ ! -f "$file1" ]; then
    echo "File $file1 does not exist."
    exit 1
fi

if [ ! -f "$file2" ]; then
    echo "File $file2 does not exist."
    exit 1
fi

# Compare the contents of the files
if cmp -s "$file1" "$file2"; then
    echo "Files $file1 and $file2 have the same content."
    rm "$file2"
    echo "So $file2 is deleted."
else
    echo "Files $file1 and $file2 have different content."
fi

```

11



```
# Check if exactly three arguments are passed
if [ $# -ne 3 ]; then
    echo "Invalid input"
    echo "Enter input in the following format: op1 operator op2"
    exit 1
fi
# Assign arguments to variables
op1=$1
operator=$2
op2=$3
# Validate if operands are numbers
if ! [[ $op1 =~ ^-?[0-9]+(\.[0-9]+)?$ ]] || ! [[ $op2 =~ ^-?[0-9]+(\.[0-9]+)?$ ]]; then
    echo "Invalid input"
    echo "Operands must be numbers."
    exit 1
fi
# Perform the arithmetic operation based on the operator
case $operator in
    +)
        result=$(echo "$op1 + $op2" | bc)
        ;;
    -)
        result=$(echo "$op1 - $op2" | bc)
        ;;
    x)
        result=$(echo "$op1 * $op2" | bc)
        ;;
    /)
        # Check for division by zero
        if [ "$op2" == "0" ]; then
            echo "Division by zero is not allowed."
            exit 1
        fi
        result=$(echo "scale=2; $op1 / $op2" | bc)
        ;;
    %)
        result=$(echo "$op1 % $op2" | bc)
        ;;
    ^)
        result=$(echo "$op1 ^ $op2" | bc)
        ;;
    *)
        echo "Invalid input"
        echo "Supported operators are: + - x / % ^"
        exit 1
        ;;
esac
# Display the result
echo "$op1 $operator $op2 = $result"
```

12