1. Write a shell script **iaop** to perform integer arithmetic on two numbers, where the value of the two numbers will be given during runtime.

**#!/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**

**# Perform arithmetic operations**

**echo "Results of arithmetic operations on $num1 and $num2:"**

**# Addition**

**sum=$((num1 + num2))**

**echo "Addition: $num1 + $num2 = $sum"**

**# Subtraction**

**diff=$((num1 - num2))**

**echo "Subtraction: $num1 - $num2 = $diff"**

**# Multiplication**

**prod=$((num1 \* num2))**

**echo "Multiplication: $num1 \* $num2 = $prod"**

**# Division (only if the second number is not zero)**

**if [ $num2 -ne 0 ]; then**

**div=$((num1 / num2))**

**echo "Division: $num1 / $num2 = $div"**

**else**

**echo "Division: Cannot divide by zero"**

**fi**

**# Modulo (remainder of the division)**

**if [ $num2 -ne 0 ]; then**

**mod=$((num1 % num2))**

**echo "Modulo: $num1 % $num2 = $mod"**

**else**

**echo "Modulo: Cannot perform modulo operation with zero"**

**fi**

gedit arithmetic\_operations.sh

chmod +x arithmetic\_operations.sh

./arithmetic\_operations.sh

1. Write a shell script **faop** to perform floating point arithmetic on two numbers, where the value of the two numbers will be given during runtime.

**#!/bin/bash**

**# Prompt the user to enter the first number (floating point)**

**echo "Enter the first number (floating point):"**

**read num1**

**# Prompt the user to enter the second number (floating point)**

**echo "Enter the second number (floating point):"**

**read num2**

**# Perform floating point arithmetic using bc**

**# Addition**

**sum=$(echo "$num1 + $num2" | bc)**

**echo "Addition: $num1 + $num2 = $sum"**

**# Subtraction**

**diff=$(echo "$num1 - $num2" | bc)**

**echo "Subtraction: $num1 - $num2 = $diff"**

**# Multiplication**

**prod=$(echo "$num1 \* $num2" | bc)**

**echo "Multiplication: $num1 \* $num2 = $prod"**

**# Division (with a check to avoid division by zero)**

**if [ $(echo "$num2 == 0" | bc) -eq 1 ]; then**

**echo "Division: Cannot divide by zero"**

**else**

**div=$(echo "scale=2; $num1 / $num2" | bc)**

**echo "Division: $num1 / $num2 = $div"**

**fi**

gedit floating\_point\_operations.sh

chmod +x floating\_point\_operations.sh

./floating\_point\_operations.sh

1. Ramesh’s basic salary is input through the keyboard. His dearness allowance is 40% of basic salary, and house rent allowance is 20% of basic salary. Write a program to calculate his gross salary.

**#!/bin/bash**

**# Prompt the user to enter Ramesh's basic salary**

**echo "Enter Ramesh's basic salary:"**

**read basic\_salary**

**# Calculate dearness allowance (40% of basic salary)**

**da=$(echo "scale=2; $basic\_salary \* 0.40" | bc)**

**# Calculate house rent allowance (20% of basic salary)**

**hra=$(echo "scale=2; $basic\_salary \* 0.20" | bc)**

**# Calculate gross salary**

**gross\_salary=$(echo "scale=2; $basic\_salary + $da + $hra" | bc)**

**# Display the results**

**echo "Basic Salary: $basic\_salary"**

**echo "Dearness Allowance (40%): $da"**

**echo "House Rent Allowance (20%): $hra"**

**echo "Gross Salary: $gross\_salary"**

1. If a five digit number is input given through the keyboard during runtime, write a program to calculate the sum of its digits.

**#!/bin/bash**

**# Prompt the user to enter a five-digit number**

**echo "Enter a five-digit number:"**

**read number**

**# Check if the input is numeric and exactly five digits long**

**if [[ $number -ge 10000 && $number -le 99999 ]]; then**

**# Initialize sum to 0**

**sum=0**

**# Loop through each digit of the number**

**while [ $number -gt 0 ]; do**

**# Extract the last digit using modulo**

**digit=$((number % 10))**

**# Add the digit to the sum**

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

**# Remove the last digit**

**number=$((number / 10))**

**done**

**# Print the result**

**echo "The sum of the digits is $sum."**

**else**

**echo "Invalid input! Please enter a valid five-digit number."**

**fi**

1. If cost price and selling price of an item is input through the keyboard, write a program to determine whether the seller has made profit or incurred loss. Also determine how much profit was made or loss incurred.

**#!/bin/bash**

**# Prompt the user to enter the cost price**

**echo "Enter the cost price of the item:"**

**read cost\_price**

**# Prompt the user to enter the selling price**

**echo "Enter the selling price of the item:"**

**read selling\_price**

**# Calculate profit or loss**

**if (( $(echo "$selling\_price > $cost\_price" | bc -l) )); then**

**profit=$(echo "$selling\_price - $cost\_price" | bc -l)**

**echo "The seller made a profit of $profit."**

**elif (( $(echo "$cost\_price > $selling\_price" | bc -l) )); then**

**loss=$(echo "$cost\_price - $selling\_price" | bc -l)**

**echo "The seller incurred a loss of $loss."**

**else**

**echo "There is no profit or loss (break-even)."**

**fi**

1. Write a shell script which receives any year from the keyboard and determines, whether the year is a leap year or not. If no argument is supplied the current year should be assumed.

**#!/bin/bash**

**# Check if a year is provided as an argument**

**if [ -z "$1" ]; then**

**# Get the current year if no argument is supplied**

**year=$(date +"%Y")**

**else**

**year=$1**

**fi**

**# Determine if the year is a leap year**

**if (( (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0) )); then**

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

**else**

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

**fi**

1. Write a shell script **allow** that will display a message to enter internal mark and percentage in attendance, if the entered mark is greater than equal to 20 and entered percentage in attendance is greater that equal to 75 then display the message Allowed for Semester otherwise display the message Not allowed.

**#!/bin/bash**

**# Prompt the user to enter internal marks**

**echo "Enter internal marks (out of 50):"**

**read internal\_marks**

**# Prompt the user to enter attendance percentage**

**echo "Enter attendance percentage:"**

**read attendance\_percentage**

**# Check if the criteria for allowance are met**

**if (( $(echo "$internal\_marks >= 20" | bc -l) && $(echo "$attendance\_percentage >= 75" | bc -l) )); then**

**echo "Allowed for Semester"**

**else**

**echo "Not Allowed for Semester"**

**fi**

1. Write a shell script **small3** that will compare three numbers passed as command line

arguments and display the smallest one.

**#!/bin/bash**

**# Prompt the user to enter three numbers**

**echo "Enter Number 1:"**

**read num1**

**echo "Enter Number 2:"**

**read num2**

**echo "Enter Number 3:"**

**read num3**

**# Determine the smallest number**

**if [ $num1 -le $num2 ] && [ $num1 -le $num3 ]; then**

**echo "Smallest: $num1"**

**elif [ $num2 -le $num1 ] && [ $num2 -le $num3 ]; then**

**echo "Smallest: $num2"**

**else**

**echo "Smallest: $num3"**

**fi**

**# Display the smallest number**

**echo "The smallest number is: $smallest"**

1. Write a shell script **check\_char** which will display one message to enter a character and according to the character entered it will display appropriate message from the following options:
   * You entered a lower case alphabet
   * You entered an upper case alphabet.
   * You have entered a digit.
   * You have entered a special symbol.
   * You have entered more than one character.

**#!/bin/bash**

**# Prompt the user to enter a character**

**echo "Enter a single character:"**

**read char**

**# Check if the input contains exactly one character**

**if [ ${#char} -ne 1 ]; then**

**echo "You have entered more than one character."**

**exit 1**

**fi**

**# Check the type of character**

**if [[ $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**

1. Write a shell script **class\_time** which will display one message to enter a day and according to the day entered it will display the DOS class time along with the room information or the message “No class on day\_name” or “Holiday” for Sunday.

**#!/bin/bash**

**# Prompt the user to enter a day**

**echo "Enter a day of the week (e.g., Monday, Tuesday, etc.):"**

**read day**

**# Convert the input to lowercase for case-insensitive comparison**

**day=$(echo "$day" | tr '[:upper:]' '[:lower:]')**

**# Determine the class time and room information**

**case $day in**

**monday)**

**echo "DOS class is at 8:00 AM in Room 1."**

**;;**

**tuesday)**

**echo "DOS class is at 9:00 AM in Room 2."**

**;;**

**wednesday)**

**echo "DOS class is at 11:00 AM in Room 3."**

**;;**

**thursday)**

**echo "DOS class is at 12:00 PM in Room 4."**

**;;**

**friday)**

**echo "DOS class is at 1:00 PM in Room 5."**

**;;**

**saturday)**

**echo "No class on Saturday."**

**;;**

**sunday)**

**echo "Holiday"**

**;;**

**\*)**

**echo "Invalid day entered. Please enter a valid day of the week."**

**;;**

**esac**

1. Write a shell script **filechk** that will take two file names as command line arguments, and check whether the content of two files are same or not . If contents of two files are same, then it will display the message: Files filename1 and filename2 have same content.

then delete the second file

and display the message: So filename2 is deleted.

Otherwise display the message: Files filename1 and filename2 have different content.

**#!/bin/bash**

**# Check if exactly two arguments are provided**

**if [ "$#" -ne 2 ]; then**

**echo "Usage: $0 filename1 filename2"**

**exit 1**

**fi**

**# Assign the 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 content of the two files**

**if cmp -s "$file1" "$file2"; then**

**echo "Files $file1 and $file2 have same content."**

**rm "$file2"**

**echo "So $file2 is deleted."**

**else**

**echo "Files $file1 and $file2 have different content."**

**fi**

gedit file1.txt (Add some content must be same as file 2)

gedit file2.txt (Add some content must be same as file 1)

gedit file3.txt (Add some different content)

gedit compare\_files.sh

chmod +x compare\_files.sh

./compare\_files.sh file1.txt file2.txt

./compare\_files.sh file1.txt file3.txt

1. Write a shell script **calculator** that will take three command line arguments, where the first argument will specify the first operand, second argument will specify the operator and the third argument will specify the second operand and display the output of the arithmetic operation specified in the following format: op1 operator op2 = result .

If the arguments will be passed in any other sequence, it will display the message: “Invalid input “

Enter input in following format: op1 operator op2

The symbols to be used for different operators are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Addition: | + | Subtraction: | - |
| Multiplication: | x | Division: | / |
| Modulo: | % | Exponent: | ^ |

**#!/bin/bash**

**# Check if exactly three arguments are passed**

**if [ "$#" -ne 3 ]; then**

**echo "Invalid input. Enter input in the following format: op1 operator op2"**

**exit 1**

**fi**

**# Assign arguments to variables**

**op1=$1**

**operator=$2**

**op2=$3**

**# Perform the calculation based on the operator**

**case $operator in**

**+)**

**result=$((op1 + op2))**

**;;**

**-)**

**result=$((op1 - op2))**

**;;**

**x)**

**result=$((op1 \* op2))**

**;;**

**/)**

**if [ "$op2" -eq 0 ]; then**

**echo "Error: Division by zero is not allowed."**

**exit 1**

**fi**

**result=$((op1 / op2))**

**;;**

**%)**

**result=$((op1 % op2))**

**;;**

**^)**

**result=$((op1 \*\* op2))**

**;;**

**\*)**

**echo "Invalid operator. Use one of the following: +, -, x, /, %, ^"**

**exit 1**

**;;**

**esac**

**# Display the result in the format: op1 operator op2 = result**

**echo "$op1 $operator $op2 = $result"**

gedit calculate.sh

chmod +x calculate.sh

./calculate.sh 5 + 3

./calculate.sh 10 – 4

./calculate.sh 6 x 2

./calculate.sh 8 / 4

./calculate.sh 10 % 3

./calculate.sh 2 ^ 3