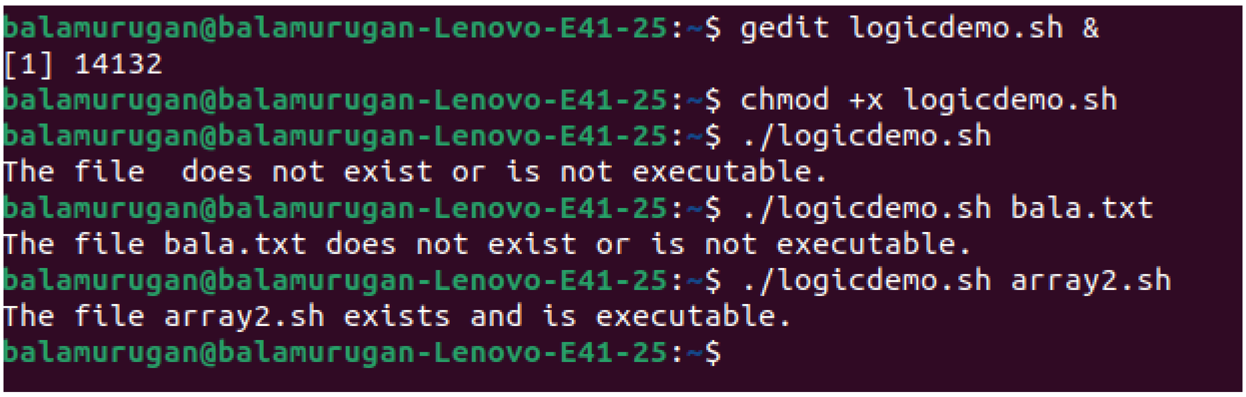
MODULE 4

Logical Operators

1) Check whether the file exists and is executable using logical operators.

```
#!/bin/bash

if [ -f "$1" ] && [ -x "$1" ]; then

  echo "The file $1 exists and is executable."

else

  echo "The file $1 does not exist or is not executable."

fi
```

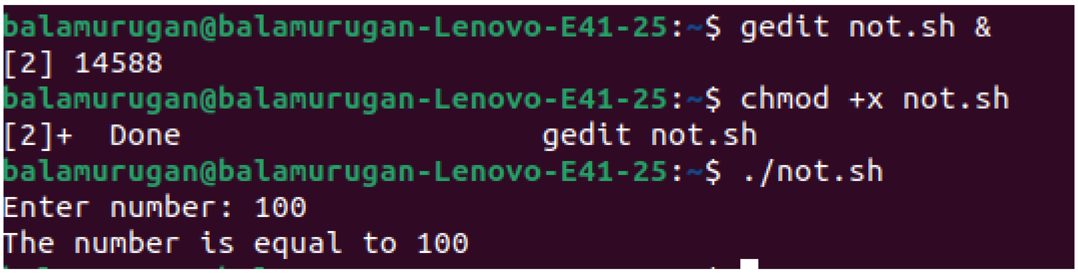


```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit logicdemo.sh &
[1] 14132
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x logicdemo.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./logicdemo.sh
The file  does not exist or is not executable.
balamurugan@balamurugan-Lenovo-E41-25:~$ ./logicdemo.sh bala.txt
The file bala.txt does not exist or is not executable.
balamurugan@balamurugan-Lenovo-E41-25:~$ ./logicdemo.sh array2.sh
The file array2.sh exists and is executable.
balamurugan@balamurugan-Lenovo-E41-25:~$
```

Arithmetic Comparison

1) Write a program to demonstrate the use of not equal to operator.
  Hint: -ne

```
  #!/bin/bash

  read -p "Enter number: " num

  if [ "$num" -ne 100 ]; then
          echo "The number is not equal to 100"
          else
          echo "The number is equal to 100"
  fi
```

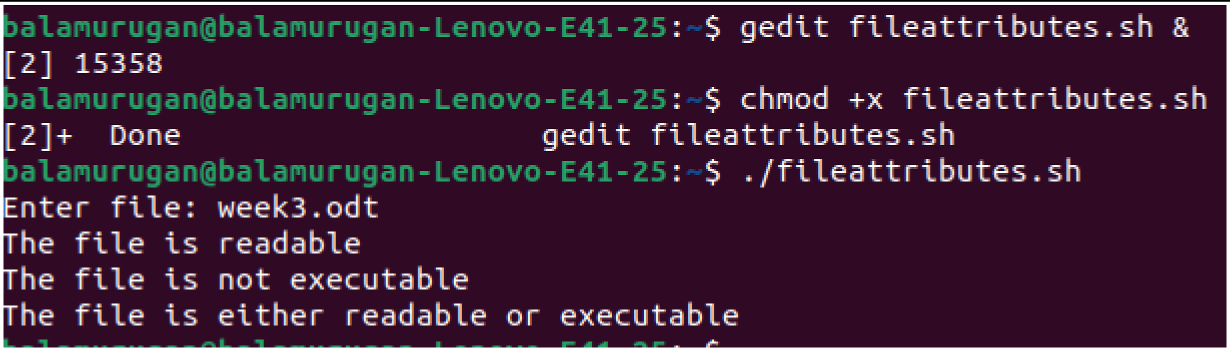


```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit not.sh &
[2] 14588
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x not.sh
[2]+  Done                    gedit not.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./not.sh
Enter number: 100
The number is equal to 100
```

String and File attributes

1) Explore some more attributes
-r
-x
-o

```
#!/bin/bash

read -p "Enter file: " file

if [ -r "$file" ]; then
        echo "The file is readable"
        else
        echo "The file is not readable"
fi

if [ -x "$file" ]; then
        echo "The file is executable"
        else
        echo "The file is not executable"
fi

if [ ! -r "$file" -o -x "$file" ]; then
        echo "The file is not readable and not executable"
        else
        echo "The file is either readable or executable"
fi
```

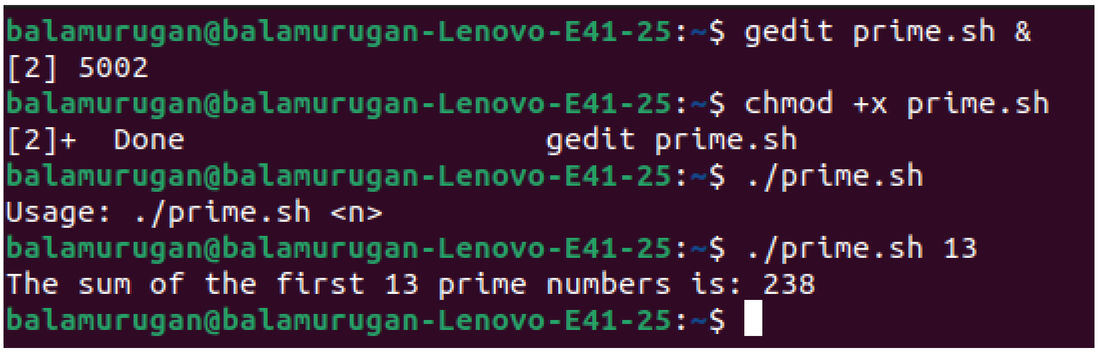


```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit fileattributes.sh &
[2] 15358
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x fileattributes.sh
[2]+  Done                    gedit fileattributes.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./fileattributes.sh
Enter file: week3.odt
The file is readable
The file is not executable
The file is either readable or executable
```

Conditional Loops

1) Find the sum of first n prime numbers.

```
#!/bin/bash

# Function to check if a number is prime
is_prime() {
  num=$1
  if [ $num -le 1 ]; then
    return 1  # Not prime
  fi
  for ((i = 2; i * i <= num; i++)); do
    if [ $((num % i)) -eq 0 ]; then
      return 1  # Not prime
    fi
  done
  return 0  # Prime
}

# Function to find the sum of the first n prime numbers
sum_of_primes() {
  n=$1
  count=0
  num=2
  sum=0

  while [ $count -lt $n ]; do
    if is_prime "$num"; then
      sum=$((sum + num))
      count=$((count + 1))
    fi
    num=$((num + 1))
  done
  echo "The sum of the first $n prime numbers is: $sum"
}

# Main program
if [ $# -eq 1 ]; then
  n="$1"
  sum_of_primes "$n"
else
  echo "Usage: $0 <n>"
fi
```

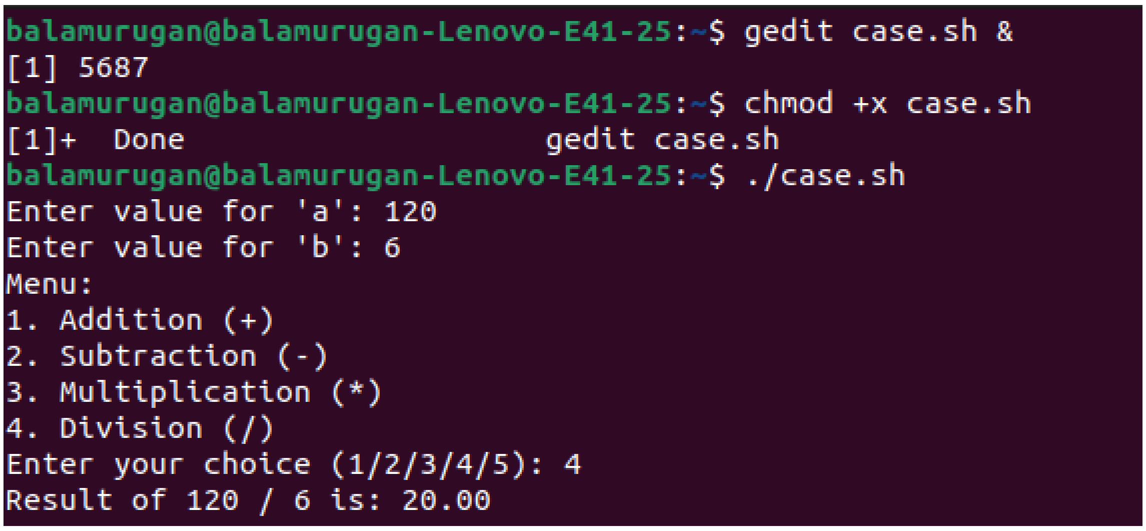


```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit prime.sh &
[2] 5002
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x prime.sh
[2]+  Done                    gedit prime.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./prime.sh
Usage: ./prime.sh <n>
balamurugan@balamurugan-Lenovo-E41-25:~$ ./prime.sh 13
The sum of the first 13 prime numbers is: 238
balamurugan@balamurugan-Lenovo-E41-25:~$ 
```

Case statement

1) Write a menu driven program for mathematical calculation
  a. It should take user inputs a and b
  b. It should ask for mathematical operator (+, -, / and *).
  c. Do the calculation
  d. Print the output

```
#!/bin/bash

# Function for addition
addition() {
  result=$((a + b))
  echo "Result of $a + $b is: $result"
}

# Function for subtraction
subtraction() {
  result=$((a - b))
  echo "Result of $a - $b is: $result"
}

# Function for multiplication
multiplication() {
  result=$((a * b))
  echo "Result of $a * $b is: $result"
}

# Function for division
division() {
  if [ "$b" -eq 0 ]; then
    echo "Error: Division by zero is not allowed."
  else
    result=$(awk "BEGIN { printf \"%.2f\", $a / $b }")
    echo "Result of $a / $b is: $result"
  fi
}

# Main menu

read -p "Enter value for 'a': " a
read -p "Enter value for 'b': " b
while true; do
  echo "Menu:"
  echo "1. Addition (+)"
  echo "2. Subtraction (-)"
  echo "3. Multiplication (*)"
  echo "4. Division (/)"

  read -p "Enter your choice (1/2/3/4/5): " choice

  case $choice in
    1)

      addition
      ;;
    2)

      subtraction
      ;;
    3)

      multiplication
      ;;
    4)

      division
      ;;
    *)
      echo "Invalid choice. Please select a valid option (1/2/3/4/5)."
      ;;
  esac
done
```

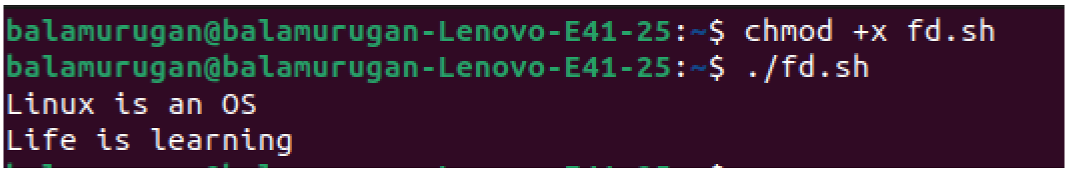


```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit case.sh &
[1] 5687
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x case.sh
[1]+  Done                    gedit case.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./case.sh
Enter value for 'a': 120
Enter value for 'b': 6
Menu:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
Enter your choice (1/2/3/4/5): 4
Result of 120 / 6 is: 20.00
```

Using File Descriptors

1) Try to append few lines to a file test.txt using file descriptor.
2) Display the content of the file using file descriptor.

```
#!/bin/bash

exec 3> test1.txt

echo "Linux is an OS" >&3
echo "Life is learning" >&3

exec 3<&-

exec 3< test1.txt
cat <&3

exec 3<&-
```



```
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x fd.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./fd.sh
Linux is an OS
Life is learning
```

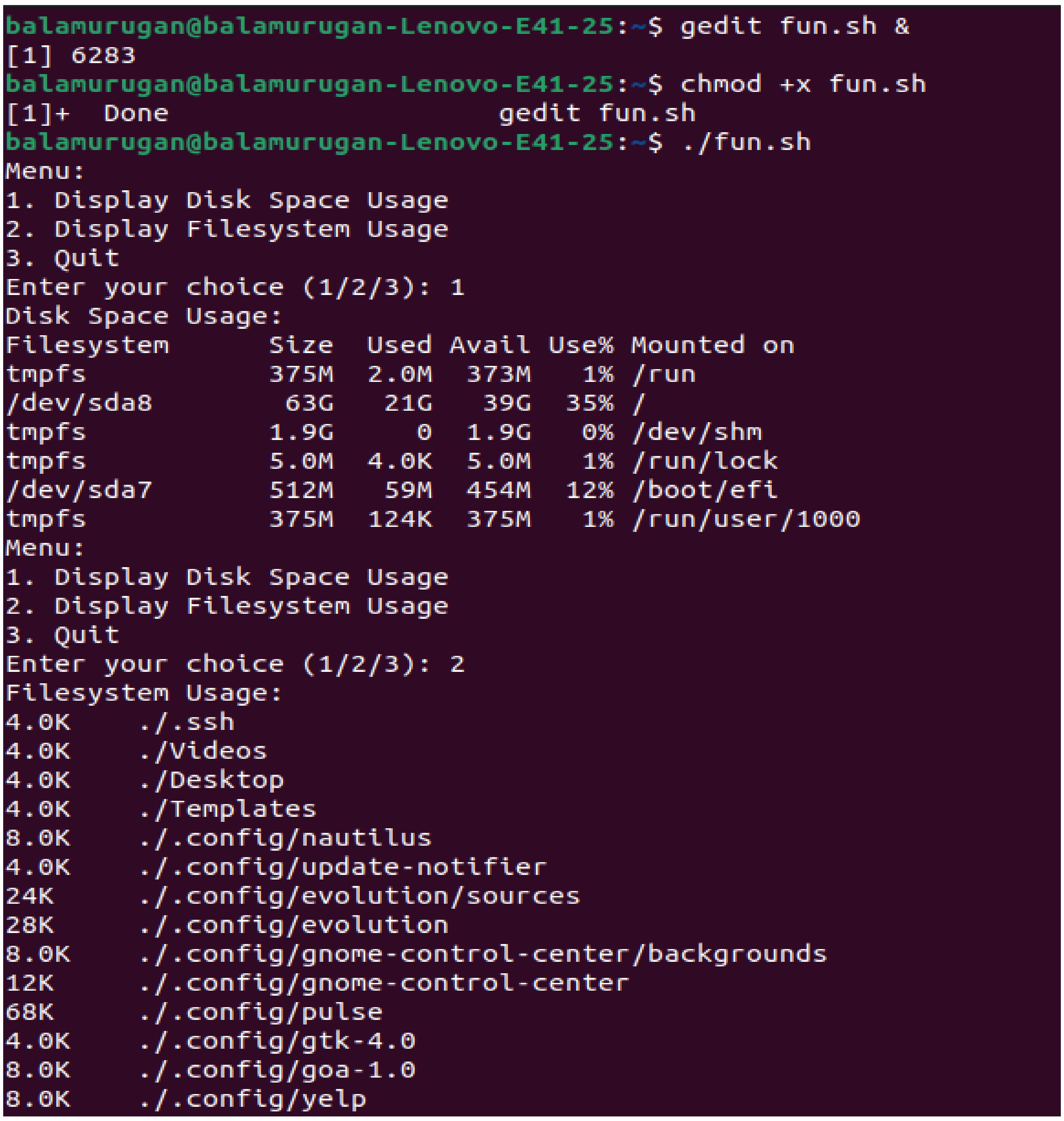
Basics of functions

1) Write a program with two functions:

  a. The first function should display diskspace usage in human readable form. (Hint: df -h)

  b. The second function should display filesystem usage in human readable form. (Hint: du -h)

```
#!/bin/bash

# Function to display disk space usage
display_disk_space() {
  echo "Disk Space Usage:"
  df -h
}

# Function to display filesystem usage
display_filesystem_usage() {
  echo "Filesystem Usage:"
  du -h
}

# Main menu
while true; do
  echo "Menu:"
  echo "1. Display Disk Space Usage"
  echo "2. Display Filesystem Usage"
  echo "3. Quit"

  read -p "Enter your choice (1/2/3): " choice

  case $choice in
    1)
      display_disk_space
      ;;
    2)
      display_filesystem_usage
      ;;
    3)
      echo "Goodbye!"
      exit 0
      ;;
    *)
      echo "Invalid choice. Please select a valid option (1/2/3)."
      ;;
  esac
done
```

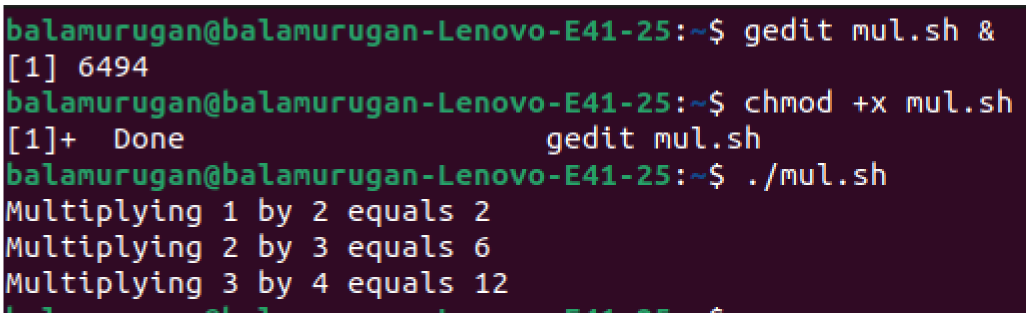


```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit fun.sh &
[1] 6283
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x fun.sh
[1]+  Done                    gedit fun.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./fun.sh
Menu:
1. Display Disk Space Usage
2. Display Filesystem Usage
3. Quit
Enter your choice (1/2/3): 1
Disk Space Usage:
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           375M  2.0M  373M   1% /run
/dev/sda8        63G   21G   39G  35% /
tmpfs           1.9G     0  1.9G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sda7       512M   59M  454M  12% /boot/efi
tmpfs           375M  124K  375M   1% /run/user/1000
Menu:
1. Display Disk Space Usage
2. Display Filesystem Usage
3. Quit
Enter your choice (1/2/3): 2
Filesystem Usage:
4.0K    ./.ssh
4.0K    ./Videos
4.0K    ./Desktop
4.0K    ./Templates
8.0K    ./.config/nautilus
4.0K    ./.config/update-notifier
24K     ./.config/evolution/sources
28K     ./.config/evolution
8.0K    ./.config/gnome-control-center/backgrounds
12K     ./.config/gnome-control-center
68K     ./.config/pulse
4.0K    ./.config/gtk-4.0
8.0K    ./.config/goa-1.0
8.0K    ./.config/yelp
```

More on functions

1) Write a program,
  a. where the function accepts two arguments.
  b. The function should multiply the two arguments.
  c. Make 3 function calls with arguments - (1, 2), (2, 3) and (3, 4)

```
#!/bin/bash

# Function to multiply two numbers
multiply() {
  local result=$(( $1 * $2 ))
  echo "Multiplying $1 by $2 equals $result"
}

# Make function calls
multiply 1 2
multiply 2 3
multiply 3 4
```



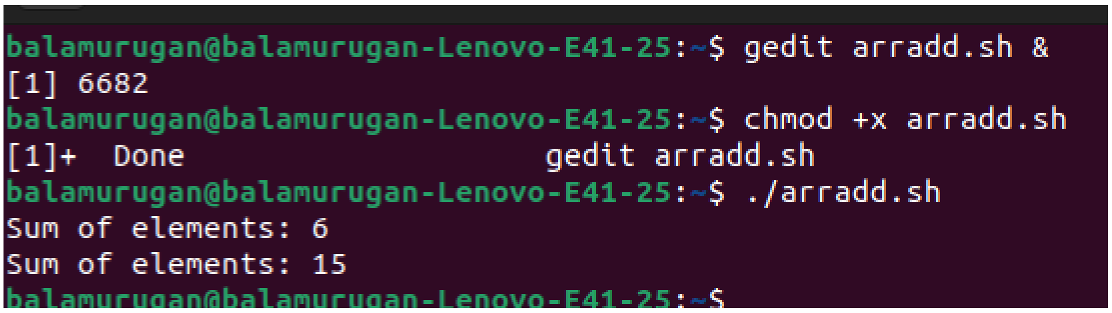
```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit mul.sh &
[1] 6494
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x mul.sh
[1]+  Done                    gedit mul.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./mul.sh
Multiplying 1 by 2 equals 2
Multiplying 2 by 3 equals 6
Multiplying 3 by 4 equals 12
```

Arrays and functions

1) Write a program,
   a. Where a function adds all the elements in an array.
   b. The function should display the sum of elements.
   c. Make 2 function calls with array elements- (1, 2, 3) and (4, 5, 6).

```
#!/bin/bash

# Function to calculate the sum of elements in an array
calculate_sum() {
  local sum=0
  for element in "${@}"; do
    sum=$((sum + element))
  done
  echo "Sum of elements: $sum"
}

# Make function calls with arrays
array1=(1 2 3)
```

```
array2=(4 5 6)

calculate_sum "${array1[@]}"
calculate_sum "${array2[@]}"
```



```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit arradd.sh &
[1] 6682
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x arradd.sh
[1]+  Done                    gedit arradd.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./arradd.sh
Sum of elements: 6
Sum of elements: 15
balamurugan@balamurugan-Lenovo-E41-25:~$
```