

```
# Print numbers from 5 to 1 using a while loop
x <- 5
while (x >= 1) {
  print(x)
  x <- x - 1
}
```

```
## [1] 5
## [1] 4
## [1] 3
## [1] 2
## [1] 1
```

```
# Print squares of numbers from 3 to 7
for (i in 3:7) {
  print(paste("The square of", i, "is", i^2))
}
```

```
## [1] "The square of 3 is 9"
## [1] "The square of 4 is 16"
## [1] "The square of 5 is 25"
## [1] "The square of 6 is 36"
## [1] "The square of 7 is 49"
```

```
# Define a number
y <- 7

# Use if-else to classify the number
if (y > 0) {
  print("Positive")
} else if (y < 0) {
  print("Negative")
} else {
  print("Zero")
}
```

```
## [1] "Positive"
```

```
# Print even numbers from 4 to 12 using a repeat loop
x <- 4
repeat {
  print(x)
  x <- x + 2
  if (x > 12) break
}
```

```
## [1] 4
## [1] 6
## [1] 8
## [1] 10
## [1] 12
```

```
# Define a vector
z <- 3:7

# Calculate squares using vectorized operation
z_squared <- z^2
```

```

print(z_squared)

## [1] 9 16 25 36 49
# Calculate the standard deviation
sd_value <- sd(c(5, 15, 25, 35, 45))
print(paste("The standard deviation is", sd_value))

## [1] "The standard deviation is 15.8113883008419"
# Calculate the product
product_value <- prod(2:6)
print(paste("The product is", product_value))

## [1] "The product is 720"
# Calculate the median
median_value <- median(c(20, 30, 40))
print(paste("The median is", median_value))

## [1] "The median is 30"
# Find the range
range_value <- range(c(10, 5, 20))
print(paste("The range is", range_value[1], "to", range_value[2]))

## [1] "The range is 5 to 20"
# Calculate the logarithm
log_value <- log(1000, base = 10)
print(paste("The logarithm value is", log_value))

## [1] "The logarithm value is 3"
# Define a function to compute the square of a number
square <- function(x) {
  return(x^2)
}

# Use the function
result <- square(4)
print(paste("The square of 4 is", result))

## [1] "The square of 4 is 16"
# Define a function to compute the sum of squares
squares_sum <- function(x, y) {
  return(x^2 + y^2)
}

# Use the function
result <- squares_sum(3, 4)
print(paste("The sum of squares of 3 and 4 is", result))

## [1] "The sum of squares of 3 and 4 is 25"
# Define a function to normalize a vector of numbers
normalize <- function(x) {
  m <- mean(x) # mean of the input
  s <- sd(x) # standard deviation of input

```

```

    return((x - m) / s) # calculate the normalized values
}

# Use the function
x <- c(20, 30, 40, 50, 60)
normalized_values <- normalize(x)
print(normalized_values)

## [1] -1.2649111 -0.6324555 0.0000000 0.6324555 1.2649111

# Define a function to normalize a vector of numbers with explicit return
normalize <- function(x) {
  m <- mean(x)
  s <- sd(x)
  normalized <- (x - m) / s
  not_normalized <- (x + m) / s
  return(normalized)
}

# Use the function
x <- c(20, 30, 40, 50, 60)
normalized_values <- normalize(x)
print(normalized_values)

## [1] -1.2649111 -0.6324555 0.0000000 0.6324555 1.2649111

# Define a function to convert temperatures
convert_temp <- function(temp, to_celsius = TRUE) {
  if (to_celsius) {
    converted <- (temp - 32) * 5/9
  } else {
    converted <- temp * 9/5 + 32
  }
  return(converted)
}

# Define a function to convert temperatures
convert_temp <- function(temp, to_celsius = TRUE) {
  if (to_celsius) {
    converted <- (temp - 32) * 5/9 # Fahrenheit to Celsius
  } else {
    converted <- temp * 9/5 + 32 # Celsius to Fahrenheit
  }
  return(converted)
}

# Use the function
temperature <- c(32, 68, 104) # Fahrenheit values
converted_temperature <- convert_temp(temperature, to_celsius = TRUE)
print("Converted to Celsius:")

## [1] "Converted to Celsius:"
print(converted_temperature)

## [1] 0 20 40

```

```
# Convert back to Fahrenheit (should match original)
converted_back <- convert_temp(converted_temperature, to_celsius = FALSE)
print("Converted back to Fahrenheit:")
```

```
## [1] "Converted back to Fahrenheit:"
```

```
print(converted_back)
```

```
## [1] 32 68 104
```