**NAME – SAARA ANAND**

**REG NO – 21BCE8156**

**SLOT – L55+L56**

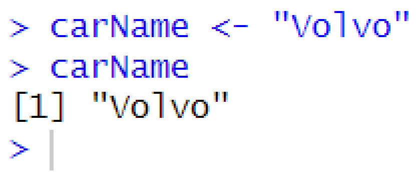
**FDA LAB ASSIGNMENT 2-**

1. **Create a variable named carName and assign the value Volvo to it.**

   CODE-

   carName <- "Volvo"

   carName



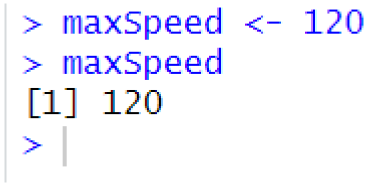
```
> carName <- "Volvo"
> carName
[1] "Volvo"
> 
```

2. **Create a variable named maxSpeed and assign the value 120 to it.**

   CODE-

   maxSpeed <- 120

   maxSpeed



```
> maxSpeed <- 120
> maxSpeed
[1] 120
> 
```

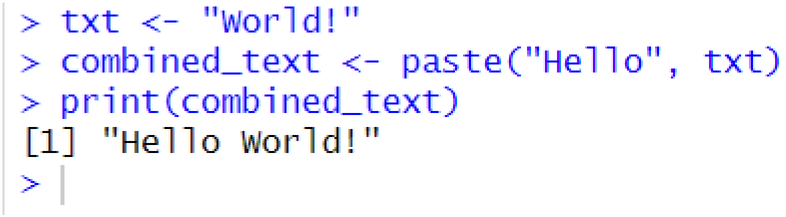
3. **Use the correct function to combine the text "Hello" with the txt variable, to output "Hello World!".**

   CODE-

   txt <- "World!"

   combined_text <- paste("Hello", txt)

print(combined_text)



```
> txt <- "World!"
> combined_text <- paste("Hello", txt)
> print(combined_text)
[1] "Hello World!"
> 
```

4. **Display the sum of 5 + 10, using two variables: x and y.**

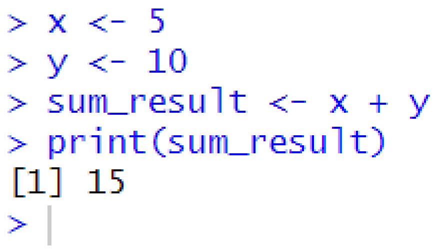
CODE-

x <- 5

y <- 10

sum_result <- x + y

print(sum_result)



```
> x <- 5
> y <- 10
> sum_result <- x + y
> print(sum_result)
[1] 15
> 
```

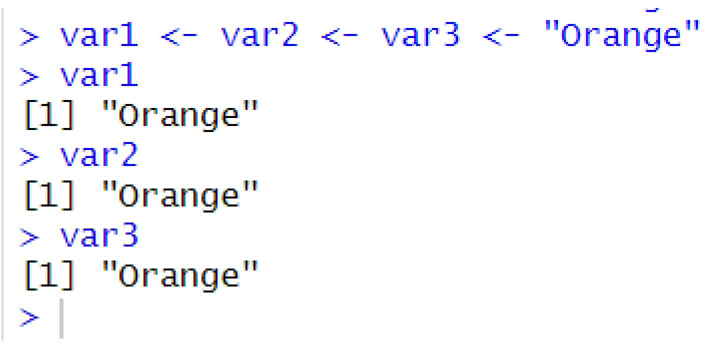
5. **Assign the value "Orange" to multiple variables in one line.**

var1 <- var2 <- var3 <- "Orange"

var1

var2

var3



```
> var1 <- var2 <- var3 <- "Orange"
> var1
[1] "Orange"
> var2
[1] "Orange"
> var3
[1] "Orange"
> 
```

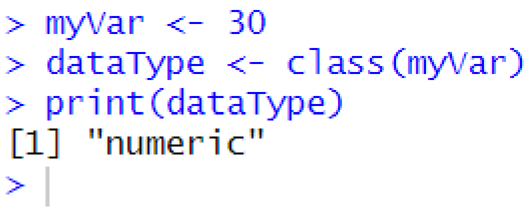
6. **myvar<30, What data type is myVar?**

   CODE-

   myVar <- 30

   dataType <- class(myVar)

   print(dataType)



```
> myVar <- 30
> dataType <- class(myVar)
> print(dataType)
[1] "numeric"
> 
```

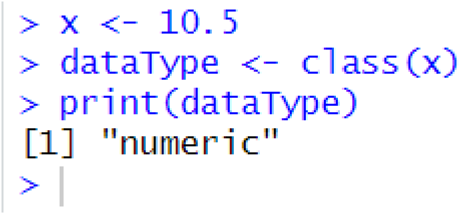
7. **x<-10.5, Which function can be used to check the data type of variable x?**

   CODE-

   x <- 10.5

   dataType <- class(x)

   print(dataType)



```
> x <- 10.5
> dataType <- class(x)
> print(dataType)
[1] "numeric"
> 
```

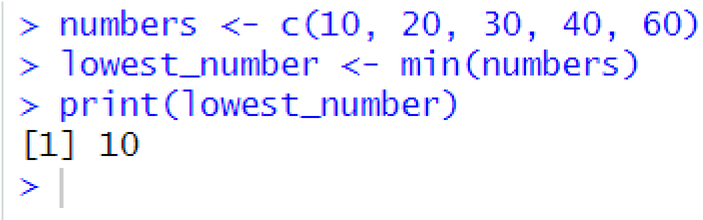
8. **Use the correct function to find the lowest number in a set (10,20,30,40,60).**

CODE-

numbers <- c(10, 20, 30, 40, 60)

lowest_number <- min(numbers)

print(lowest_number)



```
> numbers <- c(10, 20, 30, 40, 60)
> lowest_number <- min(numbers)
> print(lowest_number)
[1] 10
> 
```

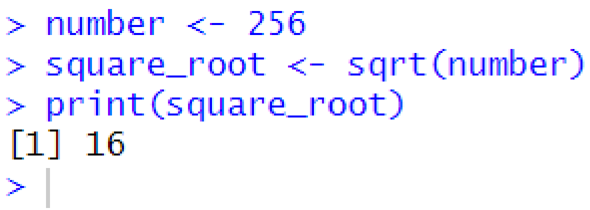
9. **Find the square root of 256.**

CODE-

number <- 256

square_root <- sqrt(number)

print(square_root)

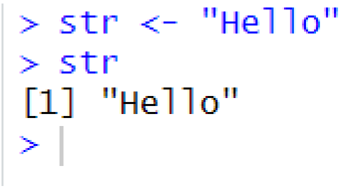


```
> number <- 256
> square_root <- sqrt(number)
> print(square_root)
[1] 16
> 
```

10. **Create a str variable with the value "Hello”.**

CODE-str <- "Hello"

str



```
> str <- "Hello"
> str
[1] "Hello"
>
```

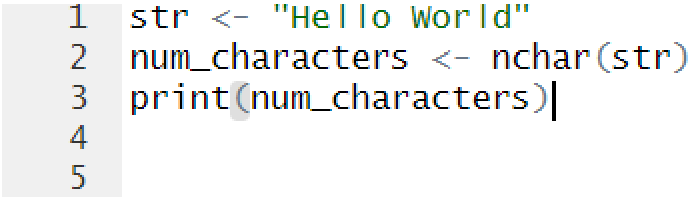
**11. Find the number of characters in the str variable, where str<-“Hello World”.**

CODE-

str <- "Hello World"

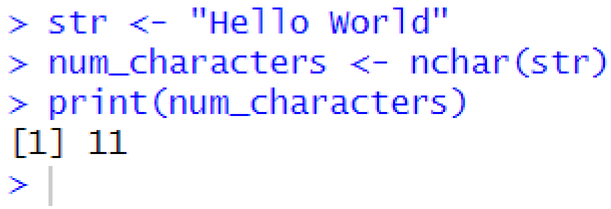
num_characters <- nchar(str)

print(num_characters)



```
str <- "Hello World"
num_characters <- nchar(str)
print(num_characters)


```



```
> str <- "Hello World"
> num_characters <- nchar(str)
> print(num_characters)
[1] 11
>
```

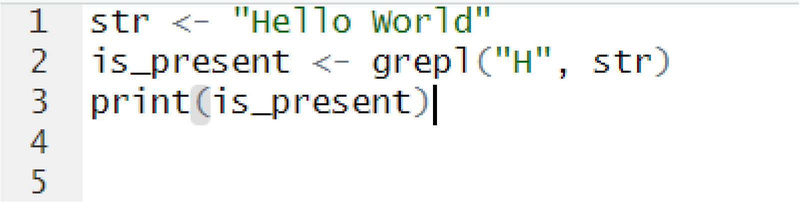
**12. Check if the character "H" is present in the str variable.**

CODE-

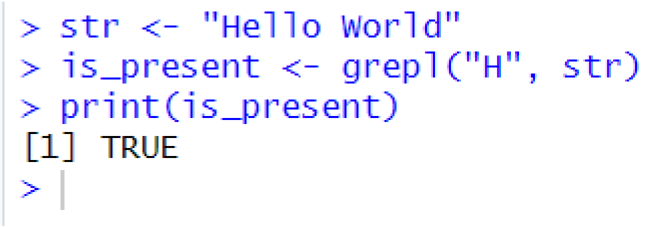
str <- "Hello World"

is_present <- grepl("H", str)

print(is_present)



```
str <- "Hello World"
is_present <- grepl("H", str)
print(is_present)
```



```
> str <- "Hello World"
> is_present <- grepl("H", str)
> print(is_present)
[1] TRUE
> 
```

**13. Combine (concatenate) the two strings :: ”VIT-AP”, “University”?**

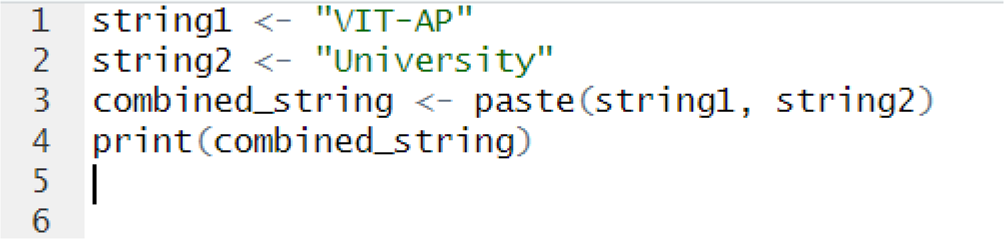
CODE-

string1 <- "VIT-AP"

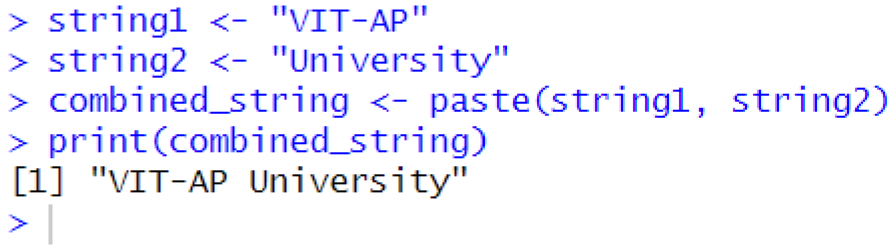
string2 <- "University"

combined_string <- paste(string1, string2)

print(combined_string)

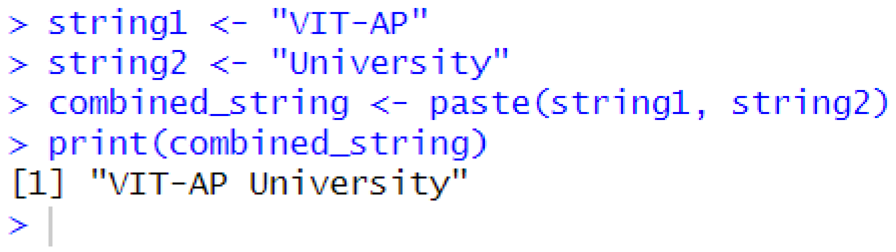


```
string1 <- "VIT-AP"
string2 <- "University"
combined_string <- paste(string1, string2)
print(combined_string)
```



```
> string1 <- "VIT-AP"
> string2 <- "University"
> combined_string <- paste(string1, string2)
> print(combined_string)
[1] "VIT-AP University"
> 
```

**14. What is the output of the following code?**



```
> string1 <- "VIT-AP"
> string2 <- "University"
> combined_string <- paste(string1, string2)
> print(combined_string)
[1] "VIT-AP University"
> 
```

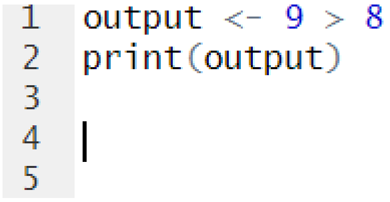
**15. What is the output of the following code?**

9>8

CODE-

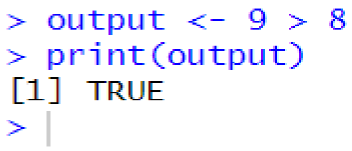
output <- 9 > 8

print(output)



```
output <- 9 > 8
print(output)


```



```
> output <- 9 > 8
> print(output)
[1] TRUE
> 
```

**16. Compare two variables and show the Boolean output.**

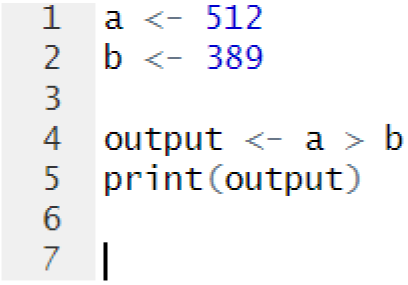
CODE-

a <- 512

b <- 389

output <- a > b

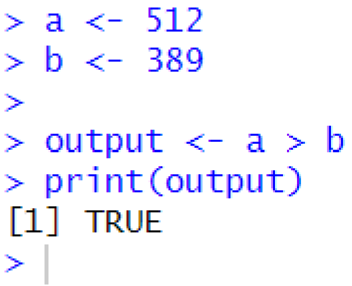
print(output)



```
a <- 512
b <- 389

output <- a > b
print(output)

```



```
> a <- 512
> b <- 389
>
> output <- a > b
> print(output)
[1] TRUE
> 
```

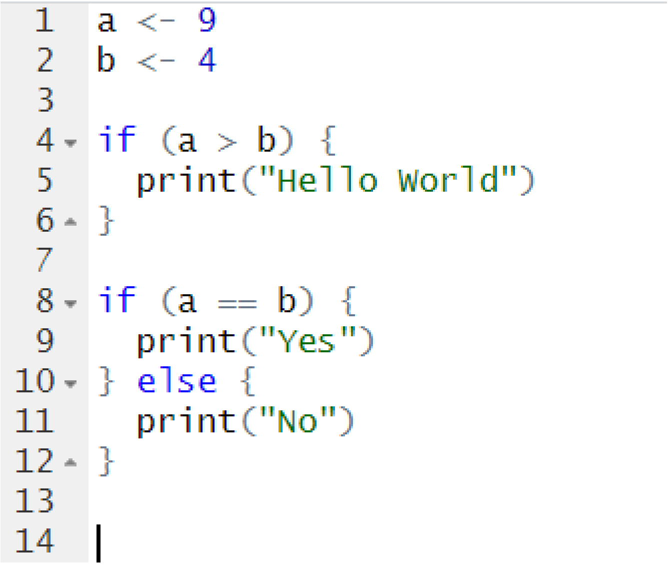
**17. Assign any value to a and b .**

**i)Print "Hello World" if a is greater than b.**

**ii) Print "Yes" if a is equal to b, otherwise print "No".**

CODE-

```
a <- 9
b <- 4
if (a > b) {
  print("Hello World")
}
if (a == b) {
  print("Yes")
} else {
  print("No")
}
```

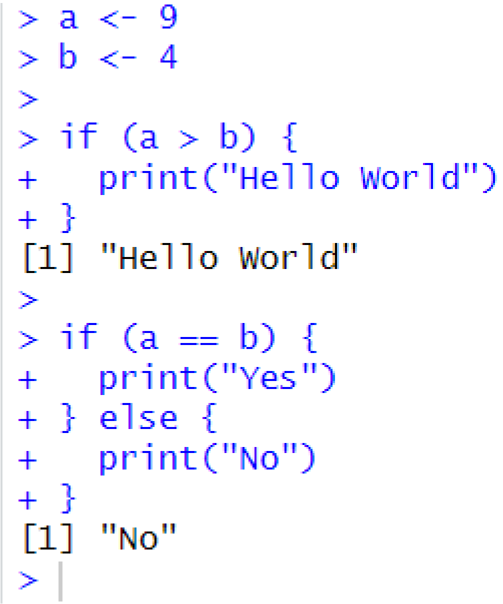


```
a <- 9
b <- 4

if (a > b) {
  print("Hello World")
}

if (a == b) {
  print("Yes")
} else {
  print("No")
}

```



```
> a <- 9
> b <- 4
>
> if (a > b) {
+   print("Hello World")
+ }
[1] "Hello World"
>
> if (a == b) {
+   print("Yes")
+ } else {
+   print("No")
+ }
[1] "No"
>
```

**18. Print i as long as i is less than 6.**

**i)In the above problem, exit the loop if i is equal to 4.**

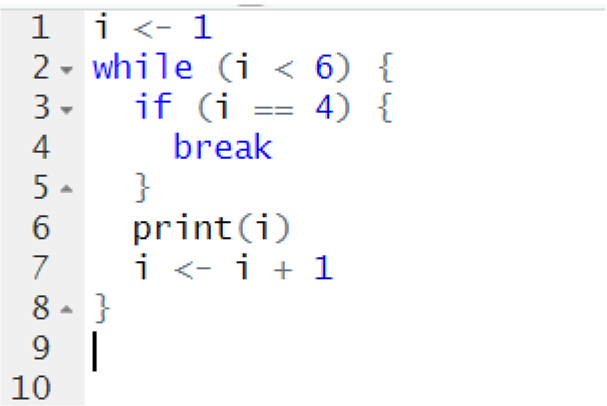
**ii) Skip the value of 3 in a loop.**

i)CODE-

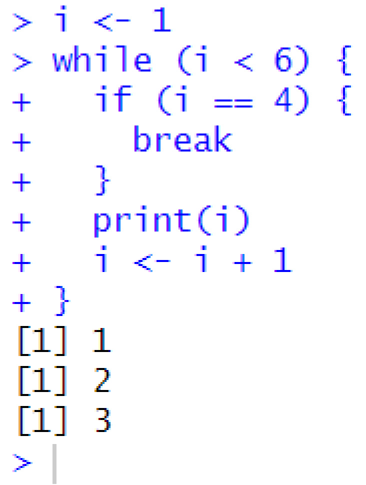
```
i <- 1

while (i < 6) {

  if (i == 4) {
    break
  }
  print(i)
  i <- i + 1
}
```



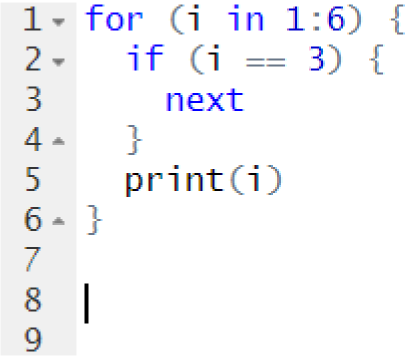
```
i <- 1
while (i < 6) {
  if (i == 4) {
    break
  }
  print(i)
  i <- i + 1
}
```



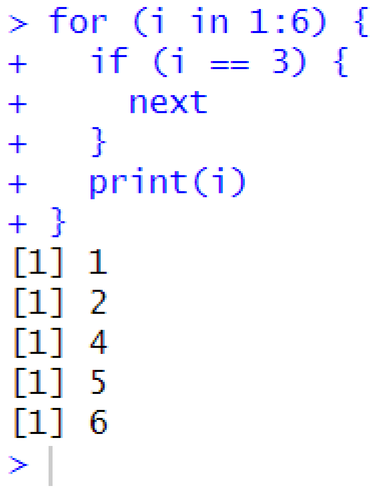
```
> i <- 1
> while (i < 6) {
+   if (i == 4) {
+     break
+   }
+   print(i)
+   i <- i + 1
+ }
[1] 1
[1] 2
[1] 3
> 
```

ii)CODE-

```
for (i in 1:6) {
  if (i == 3) {
    next
  }
  print(i)
}
```



```
for (i in 1:6) {
  if (i == 3) {
    next
  }
  print(i)
}
```

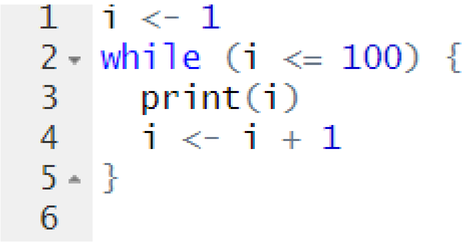


```
> for (i in 1:6) {
+   if (i == 3) {
+     next
+   }
+   print(i)
+ }
[1] 1
[1] 2
[1] 4
[1] 5
[1] 6
> 
```

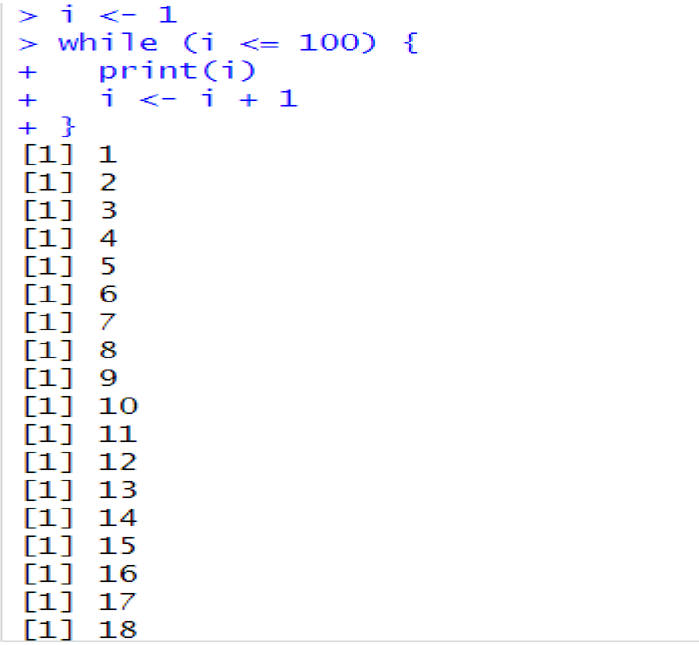
**19. Iterate over a sequence of numbers 1to 100 and print each number.**

CODE-

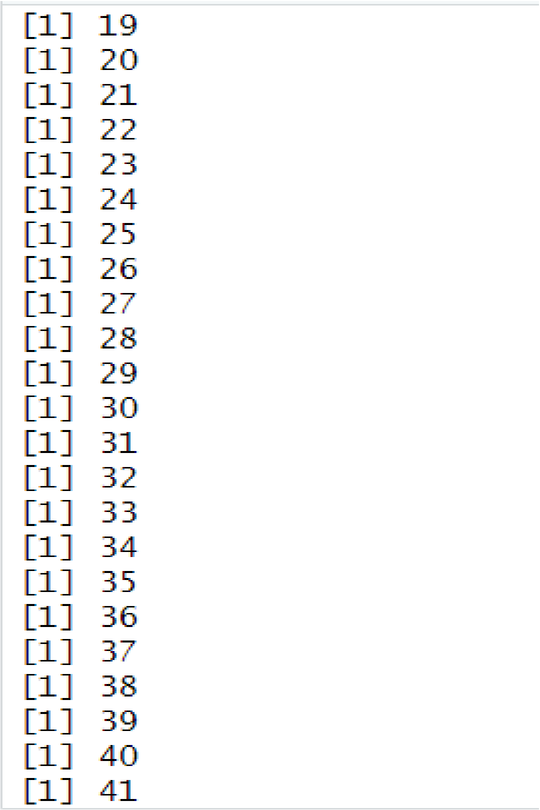
```
i <- 1
while (i <= 100) {
  print(i)
  i <- i + 1
}
```



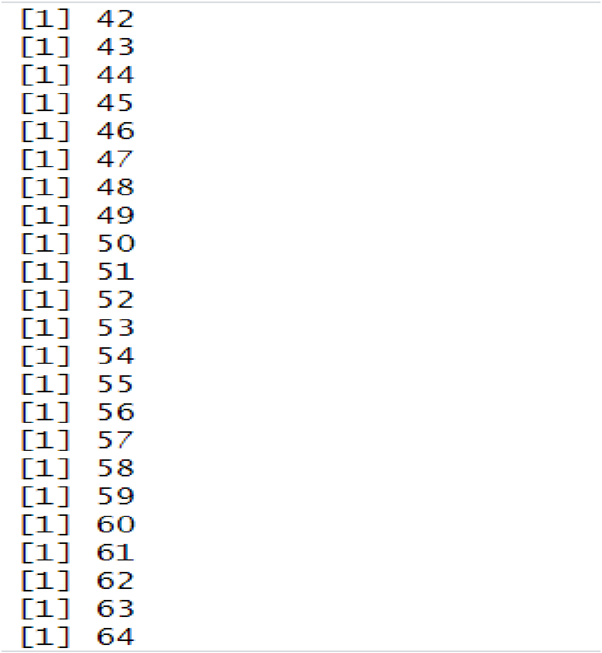
```
i <- 1
while (i <= 100) {
  print(i)
  i <- i + 1
}
```



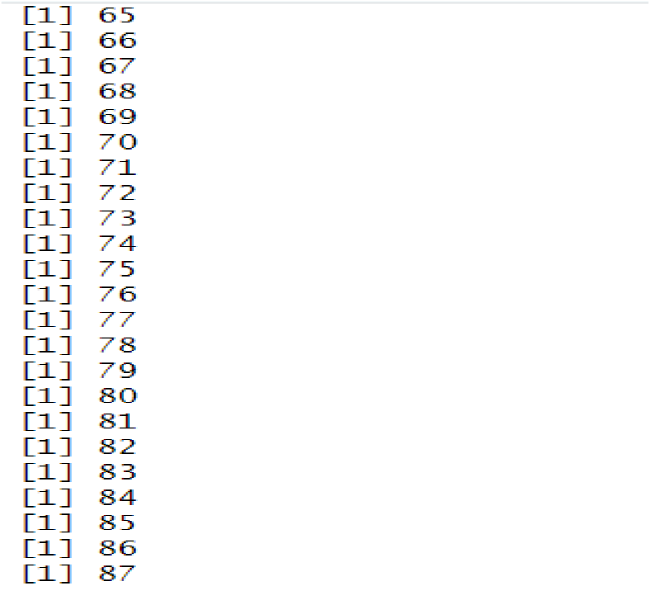
```
> i <- 1
> while (i <= 100) {
+   print(i)
+   i <- i + 1
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
[1] 16
[1] 17
[1] 18
```



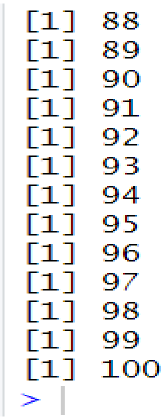
```
[1] 19
[1] 20
[1] 21
[1] 22
[1] 23
[1] 24
[1] 25
[1] 26
[1] 27
[1] 28
[1] 29
[1] 30
[1] 31
[1] 32
[1] 33
[1] 34
[1] 35
[1] 36
[1] 37
[1] 38
[1] 39
[1] 40
[1] 41
```



```
[1] 42
[1] 43
[1] 44
[1] 45
[1] 46
[1] 47
[1] 48
[1] 49
[1] 50
[1] 51
[1] 52
[1] 53
[1] 54
[1] 55
[1] 56
[1] 57
[1] 58
[1] 59
[1] 60
[1] 61
[1] 62
[1] 63
[1] 64
```



```
[1] 65
[1] 66
[1] 67
[1] 68
[1] 69
[1] 70
[1] 71
[1] 72
[1] 73
[1] 74
[1] 75
[1] 76
[1] 77
[1] 78
[1] 79
[1] 80
[1] 81
[1] 82
[1] 83
[1] 84
[1] 85
[1] 86
[1] 87
```

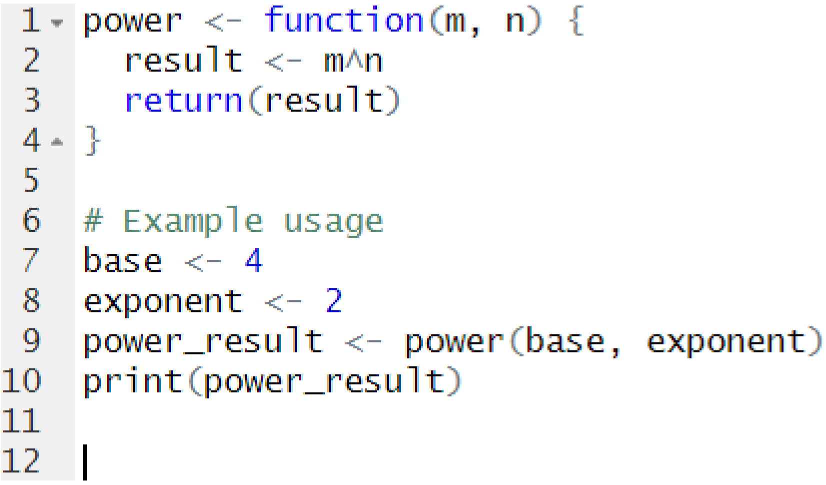


```
[1] 88
[1] 89
[1] 90
[1] 91
[1] 92
[1] 93
[1] 94
[1] 95
[1] 96
[1] 97
[1] 98
[1] 99
[1] 100
> 
```

**20. Write a function to find m^n.**

CODE-

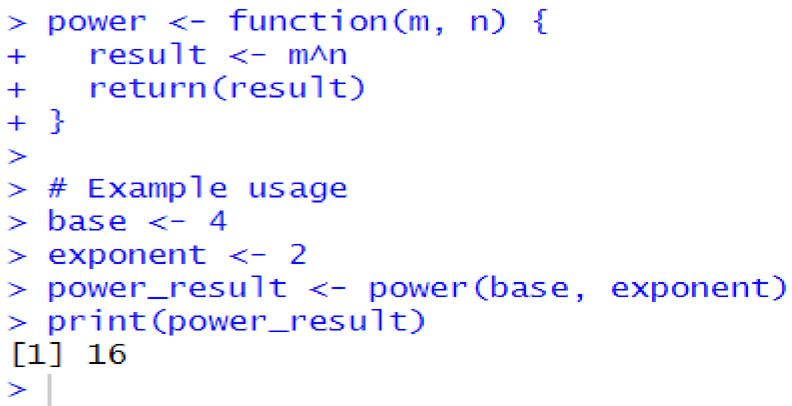
```
power <- function(m, n) {
  result <- m^n
  return(result)
}
# Example usage
base <- 2
exponent <- 3
power_result <- power(base, exponent)
print(power_result)
```



```
power <- function(m, n) {
  result <- m^n
  return(result)
}

# Example usage
base <- 4
exponent <- 2
power_result <- power(base, exponent)
print(power_result)

```

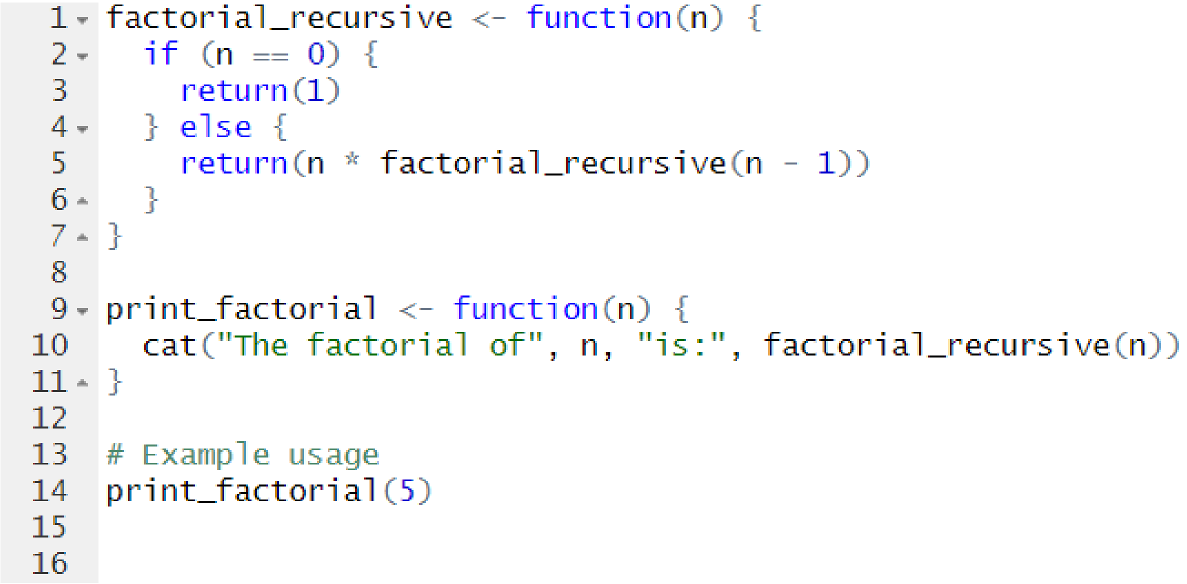


```
> power <- function(m, n) {
+   result <- m^n
+   return(result)
+ }
>
> # Example usage
> base <- 4
> exponent <- 2
> power_result <- power(base, exponent)
> print(power_result)
[1] 16
>
```

**21. Write a recursive function to print the factorial of a given number.**

CODE-

```
factorial_recursive <- function(n) {
 if (n == 0) {
   return(1)
 } else {
   return(n * factorial_recursive(n - 1))
 }
}
print_factorial <- function(n) {
 cat("The factorial of", n, "is:", factorial_recursive(n))
}
# Example usage
print_factorial(5)
```

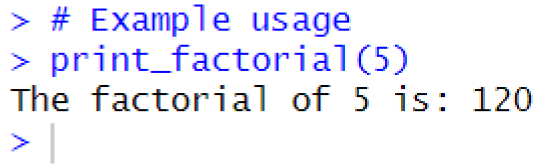


```
factorial_recursive <- function(n) {
  if (n == 0) {
    return(1)
  } else {
    return(n * factorial_recursive(n - 1))
  }
}

print_factorial <- function(n) {
  cat("The factorial of", n, "is:", factorial_recursive(n))
}

# Example usage
print_factorial(5)


```



```
> # Example usage
> print_factorial(5)
The factorial of 5 is: 120
> 
```

**-----------------------------------------------X-----------------------------------------------**