



Name: Muhammad Azlan Shah

Roll No: 26112

Section: CA

Course: Programming For AI

Teaher: Mam Zara Zainab

Assignment No 3 - Programming for AI

Task:01 with code and output given Below:

1. Variables

Create variables to store your name, age, and GPA.

Print all variables using the print() function.

A screenshot of the RStudio interface. On the left, the 'R Script' pane shows the following R code:

```
1 # Name: Muhammad Azlan Shah
2 # Roll No: 26112
3 # Task 1: Variables
4
5 name <- "Muhammad Azlan Shah"
6 age <- 20
7 gpa <- 3.5
8
9 print(name)
10 print(age)
11 print(gpa)
12
13 |
```

On the right, the 'Global Environment' pane shows the variables defined:

Values
age 20
gpa 3.5
name "Muhammad Azlan Shah"

Below the script pane, the 'Console' tab is active, showing the same R code and its corresponding output:

```
> # Name: Muhammad Azlan Shah
> # Roll No: 26112
> # Task 1: Variables
>
> name <- "Muhammad Azlan Shah"
> age <- 20
> gpa <- 3.5
>
> print(name)
[1] "Muhammad Azlan Shah"
> print(age)
[1] 20
> print(gpa)
[1] 3.5
```

Task:02 with code and output given Below:

2. Data Types

Create examples of numeric, character, logical, and vector data types.
Use class() to display the data type of each variable.

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Shows an R script with code defining variables and their classes.
- Console:** Shows the execution of the script, displaying the values assigned to each variable and their corresponding data types.
- Environment View:** Shows a table of variable names and their values.

```

1 # Name: Muhammad Azlan Shah
2 # Roll No: 26112
3 # Task 2: Data Types
4
5 num <- 25.5
6 char <- "R Programming"
7 logic <- TRUE
8 vec <- c(1, 2, 3, 4, 5)
9
10 class(num)
11 class(char)
12 class(logic)
13 class(vec)
14
14:1 [Top Level] : R Script

```

```

> num <- 25.5
> char <- "R Programming"
> logic <- TRUE
> vec <- c(1, 2, 3, 4, 5)
>
> class(num)
[1] "numeric"
> class(char)
[1] "character"
> class(logic)
[1] "logical"
> class(vec)
[1] "numeric"
>

```

Variables	Values
age	20
char	"R Programming"
gpa	3.5
logic	TRUE
name	"Muhammad Azlan Shah"
num	25.5
vec	num [1:5] 1 2 3 4 5

Task:03 with code and output given Below:

Operators

Perform arithmetic operations (+, -, *, /, ^) on two numbers.
Use relational and logical operators with suitable examples

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Shows an R script demonstrating various operators: arithmetic, relational, and logical.

```

1 # Name: Muhammad Azlan Shah
2 # Roll No: 26112
3 # Task 3: Operators
4
5 a <- 10
6 b <- 5
7
8 # Arithmetic Operators
9 a + b
10 a - b
11 a * b
12 a / b
13 a ^ b
14
15 # Relational Operators
16 a > b
17 a < b
18 a == b
19
20 # Logical Operators
21 (a > b) & (b < 10)
22 (a > b) | (b > 10)
23
24

```

Output is given Below:

```
Console | Background Jobs X
R 4.5.2 · ~/ |
> a <- 10
> b <- 5
> # Arithmetic Operators
> a + b
[1] 15
> a - b
[1] 5
> a * b
[1] 50
> a / b
[1] 2
> a ^ b
[1] 1e+05
> # Relational Operators
> a > b
[1] TRUE
> a < b
[1] FALSE
> a == b
[1] FALSE
> # Logical operators
> (a > b) & (b < 10)
[1] TRUE
> (a > b) | (b > 10)
[1] TRUE
> |
```

Task:04 with code and output given Below:

if...else Statement:

Part:01

The screenshot shows the RStudio interface. The top panel displays an R script with the following code:

```
1 # Name: Muhammad Azlan Shah
2 # Roll No: 26112
3 # Task 4: if-else (Positive / Negative / zero)
4
5 num <- -5
6
7 if (num > 0) {
8   print("Number is Positive")
9 } else if (num < 0) {
10  print("Number is Negative")
11 } else {
12  print("Number is zero")
13 }
```

The bottom panel shows the R console output:

```
10:30 (Top Level) R Script
Console Background Jobs
R 4.5.2 . ~/ ↵
>
> num <- -5
> if (num > 0) {
+   print("Number is Positive")
+ } else if (num < 0) {
+   print("Number is Negative")
+ } else {
+   print("Number is zero")
+ }
[1] "Number is Negative"
> |
```

Part:02

The screenshot shows the RStudio interface. The top panel displays an R script with the following code:

```
1 # Name: Muhammad Azlan Shah
2 # Roll No: 26112
3 # Task 4: if-else (Pass / Fail)
4
5 marks <- 65
6
7 if (marks >= 50) {
8   print("student has Passed")
9 } else {
10  print("student has Failed")
11 }
```

The bottom panel shows the R console output:

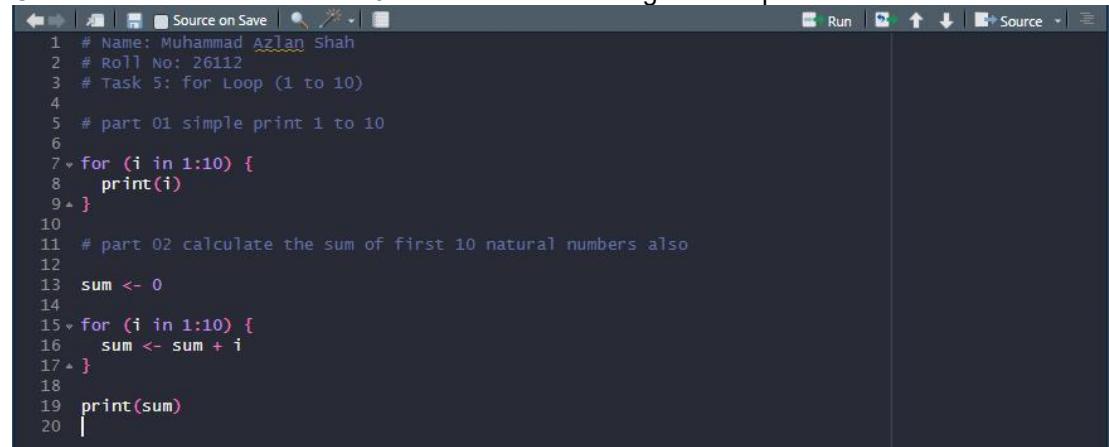
```
13:1 (Top Level) R Script
Console Background Jobs
R 4.5.2 . ~/ ↵
> # Roll No: 26112
> # Task 4: if-else (Pass / Fail)
>
> marks <- 65
>
> if (marks >= 50) {
+   print("student has Passed")
+ } else {
+   print("student has Failed")
+ }
[1] "student has Passed"
> |
```

Task:05 with code and output given Below:

For Loop:

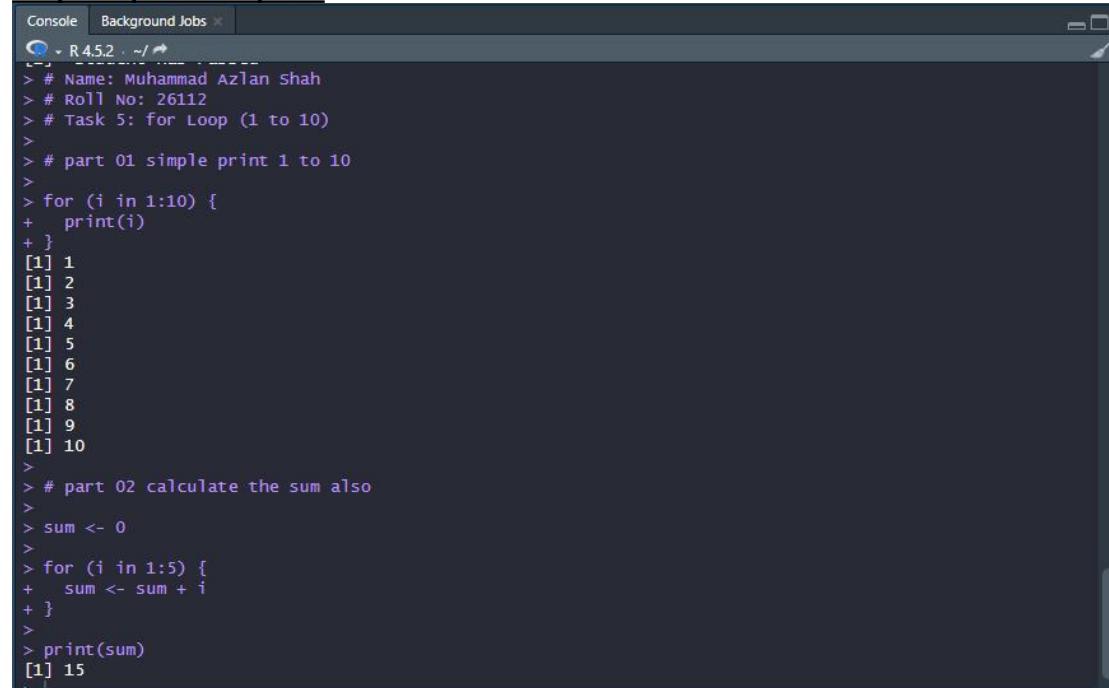
Use a for loop to print numbers from 1 to 10.

Calculate the sum of the first 10 natural numbers using a for loop.



```
1 # Name: Muhammad Azlan Shah
2 # Roll No: 26112
3 # Task 5: for Loop (1 to 10)
4
5 # part 01 simple print 1 to 10
6
7 for (i in 1:10) {
8   print(i)
9 }
10
11 # part 02 calculate the sum of first 10 natural numbers also
12
13 sum <- 0
14
15 for (i in 1:10) {
16   sum <- sum + i
17 }
18
19 print(sum)
20
```

Output of part 1 and part 2



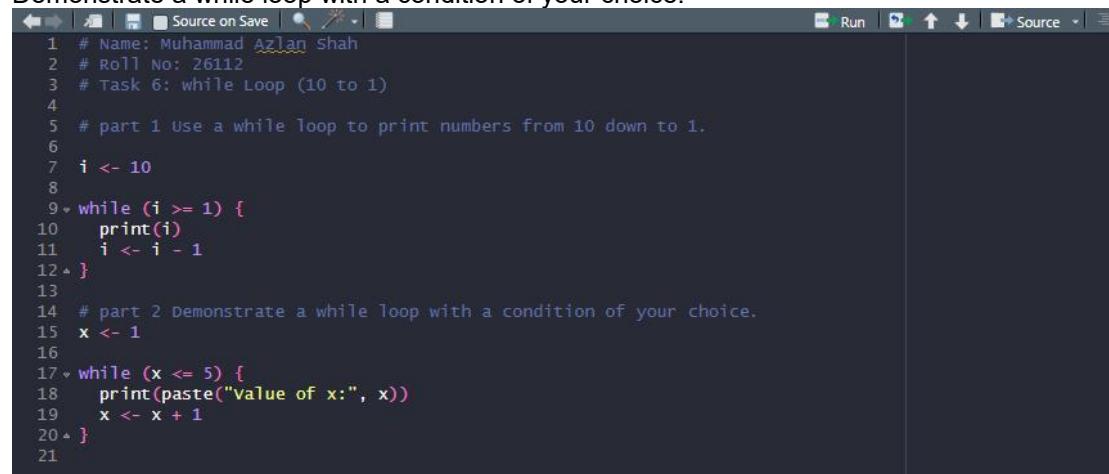
```
Console Background Jobs ✎
R 4.5.2 . ~/r
> # Name: Muhammad Azlan Shah
> # Roll No: 26112
> # Task 5: for Loop (1 to 10)
>
> # part 01 simple print 1 to 10
>
> for (i in 1:10) {
+   print(i)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
>
> # part 02 calculate the sum also
>
> sum <- 0
>
> for (i in 1:5) {
+   sum <- sum + i
+ }
>
> print(sum)
[1] 15
```

Task:06 with code and output given Below:

while Loop:

Use a while loop to print numbers from 10 down to 1.

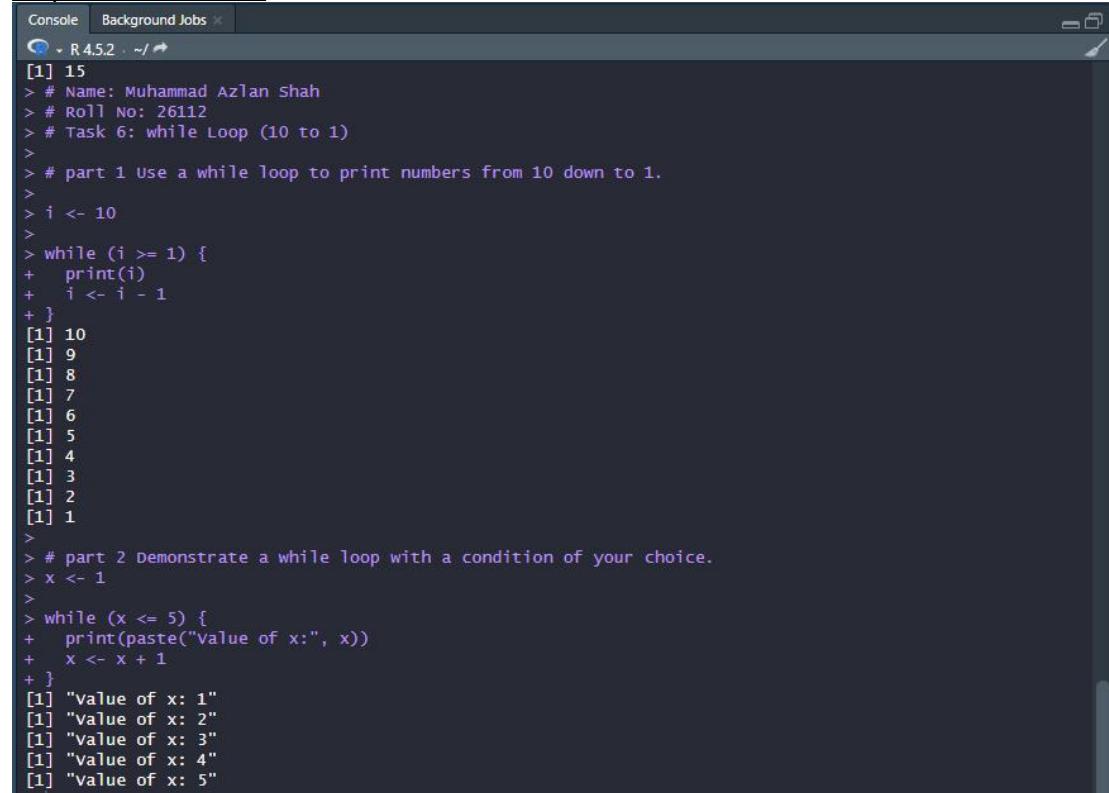
Demonstrate a while loop with a condition of your choice.



The screenshot shows the RStudio interface with the code for Task 06. The code consists of two parts. Part 1 uses a while loop to print numbers from 10 down to 1. Part 2 demonstrates a while loop with a condition of the user's choice, printing values of x from 1 to 5. The code is as follows:

```
1 # Name: Muhammad Azlan Shah
2 # Roll No: 26112
3 # Task 6: while Loop (10 to 1)
4
5 # part 1 Use a while loop to print numbers from 10 down to 1.
6
7 i <- 10
8
9 while (i >= 1) {
10   print(i)
11   i <- i - 1
12 }
13
14 # part 2 Demonstrate a while loop with a condition of your choice.
15 x <- 1
16
17 while (x <= 5) {
18   print(paste("value of x:", x))
19   x <- x + 1
20 }
21
```

Output is Given Below:



The screenshot shows the RStudio console output for the code. It starts with the R version (R 4.5.2) and the current working directory (~). The code is run, and the output shows the numbers from 10 down to 1, followed by the values of x from 1 to 5. The output is as follows:

```
Console Background Jobs ×
R 4.5.2 . ~/🔗
[1] 15
> # Name: Muhammad Azlan Shah
> # Roll No: 26112
> # Task 6: while Loop (10 to 1)
>
> # part 1 Use a while loop to print numbers from 10 down to 1.
>
> i <- 10
>
> while (i >= 1) {
+   print(i)
+   i <- i - 1
+
[1] 10
[1] 9
[1] 8
[1] 7
[1] 6
[1] 5
[1] 4
[1] 3
[1] 2
[1] 1
>
> # part 2 Demonstrate a while loop with a condition of your choice.
> x <- 1
>
> while (x <= 5) {
+   print(paste("value of x:", x))
+   x <- x + 1
+
[1] "value of x: 1"
[1] "value of x: 2"
[1] "value of x: 3"
[1] "value of x: 4"
[1] "value of x: 5"
```

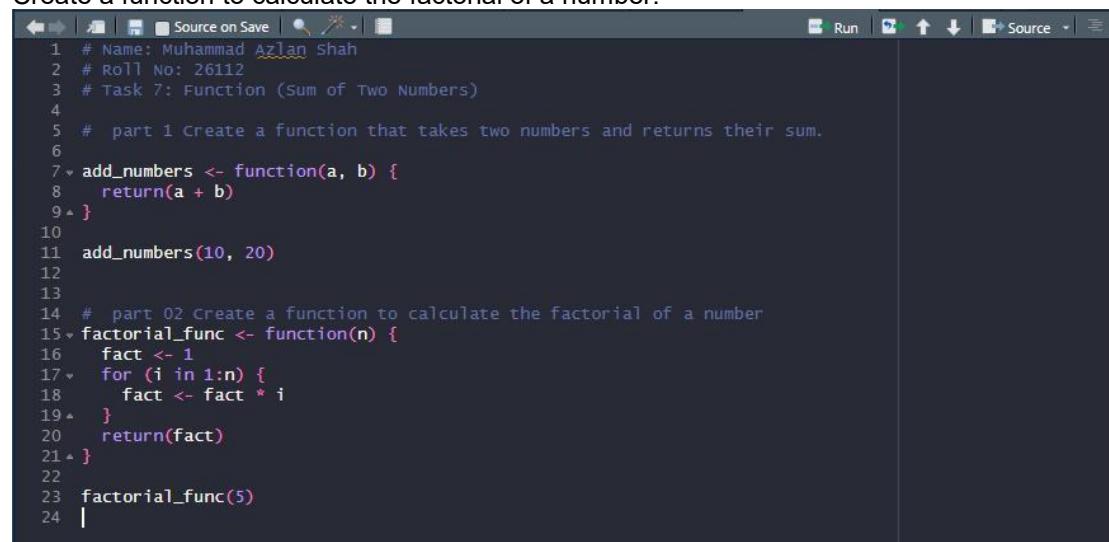
Task:07 with code and output given Below:

while Loop:

Functions

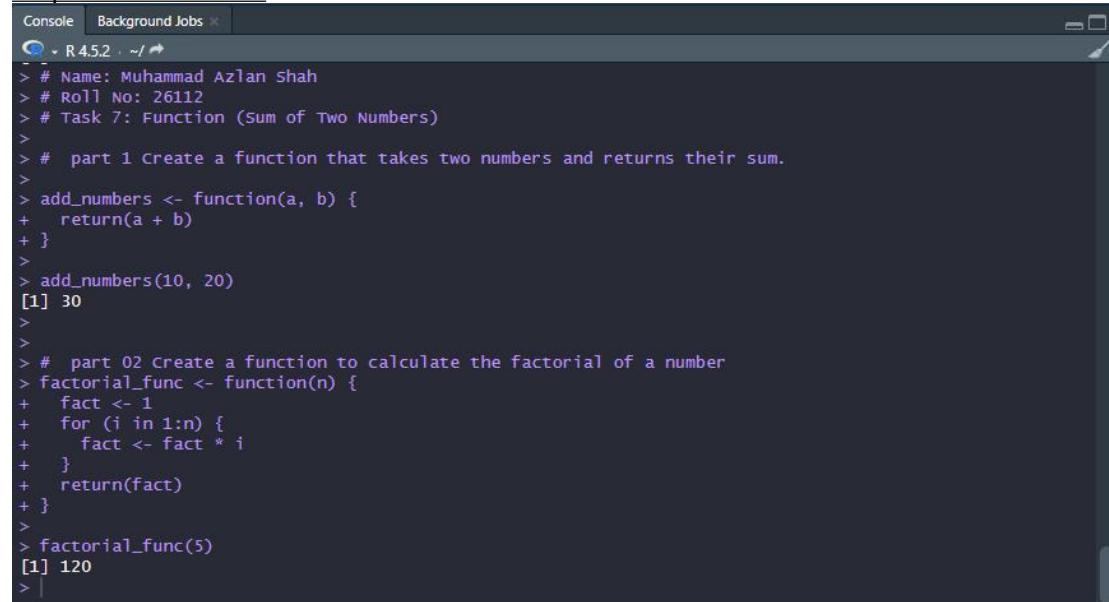
Create a function that takes two numbers and returns their sum.

Create a function to calculate the factorial of a number.



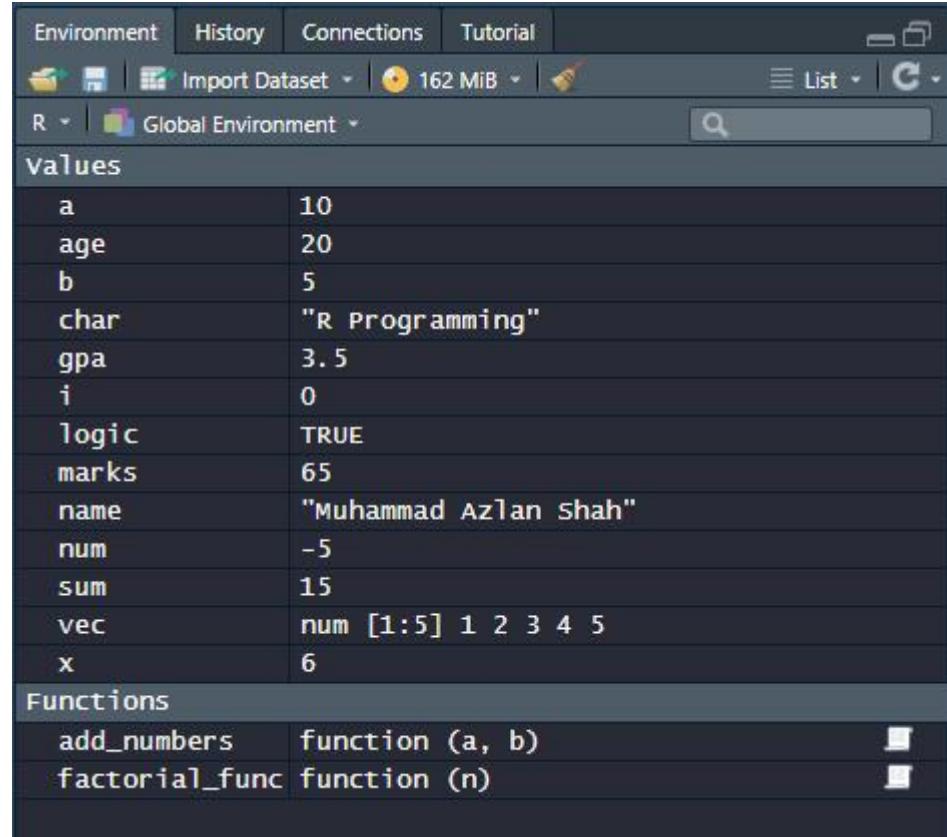
```
1 # Name: Muhammad Azlan Shah
2 # Roll No: 26112
3 # Task 7: Function (sum of Two Numbers)
4
5 # part 01 Create a function that takes two numbers and returns their sum.
6
7 add_numbers <- function(a, b) {
8   return(a + b)
9 }
10
11 add_numbers(10, 20)
12
13
14 # part 02 Create a function to calculate the factorial of a number
15 factorial_func <- function(n) {
16   fact <- 1
17   for (i in 1:n) {
18     fact <- fact * i
19   }
20   return(fact)
21 }
22
23 factorial_func(5)
24 |
```

Output is Given Below:



```
Console Background Jobs ×
R 4.5.2 ~/ ↵
> # Name: Muhammad Azlan Shah
> # Roll No: 26112
> # Task 7: Function (Sum of Two Numbers)
>
> # part 01 Create a function that takes two numbers and returns their sum.
>
> add_numbers <- function(a, b) {
+   return(a + b)
+ }
>
> add_numbers(10, 20)
[1] 30
>
>
> # part 02 Create a function to calculate the factorial of a number
> factorial_func <- function(n) {
+   fact <- 1
+   for (i in 1:n) {
+     fact <- fact * i
+   }
+   return(fact)
+ }
>
> factorial_func(5)
[1] 120
> |
```

IN THE END THE GLOBAL ENVIRONMENT SCREEN SHOT IS GIVEN BELOW OF R STUDIO:



The screenshot shows the RStudio interface with the Global Environment tab selected. The top menu bar includes Environment, History, Connections, Tutorial, and several icons. The Global Environment pane displays two sections: 'values' and 'Functions'. The 'values' section lists various R objects with their values:

Object	Type	Value
a		10
age		20
b		5
char		"R Programming"
gpa		3.5
i		0
logic		TRUE
marks		65
name		"Muhammad Azlan Shah"
num		-5
sum		15
vec		num [1:5] 1 2 3 4 5
x		6

The 'Functions' section lists two user-defined functions:

Name	Definition
add_numbers	function (a, b)
factorial_func	function (n)

The Global Environment displays all variables and functions created during the execution of the R script, confirming that the code was successfully run and the objects were stored in memory.

LIKE VAR, FUNC, OBJECT, VECTORS, LOOP COUNTERS ETC....