**PRACTICAL 8**

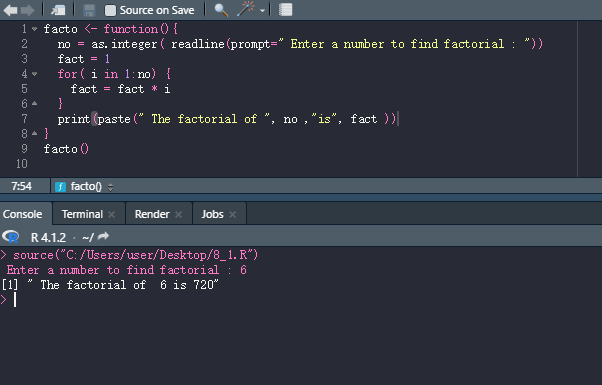
Write R Programs for following using if/else and loops constructs.

# **1.} Program To Find the Factorial of a Number.**

Algorithm

1. Step 1: Start
2. Step 2: Declare Variable n, fact, i
3. Step 3: Read number from User
4. Step 4: Initialize Variable fact=1 and i=1
5. Step 5: Repeat Until i<=number
6. 5.1 fact=fact\*i
7. 5.2 i=i+1
8. Step 6: Print fact
9. Step 7: Stop

Solution:



**2.Program To Check whether the number is odd or even.**

Algorithm:

READ number

remainder=number%2

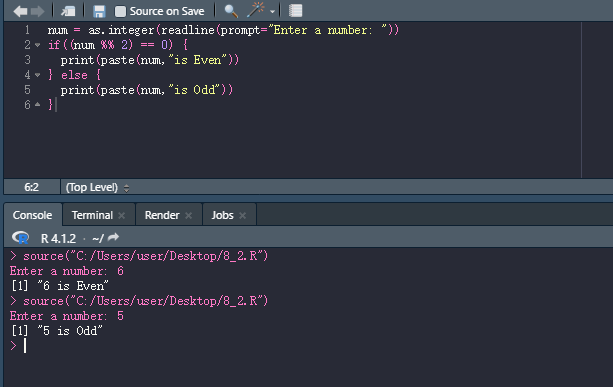
IF remainder==0

WRITE "Even Number"

ELSE

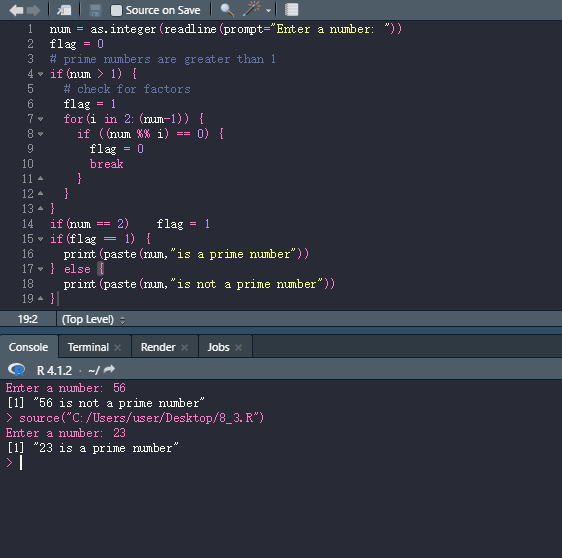
WRITE "Odd Number"

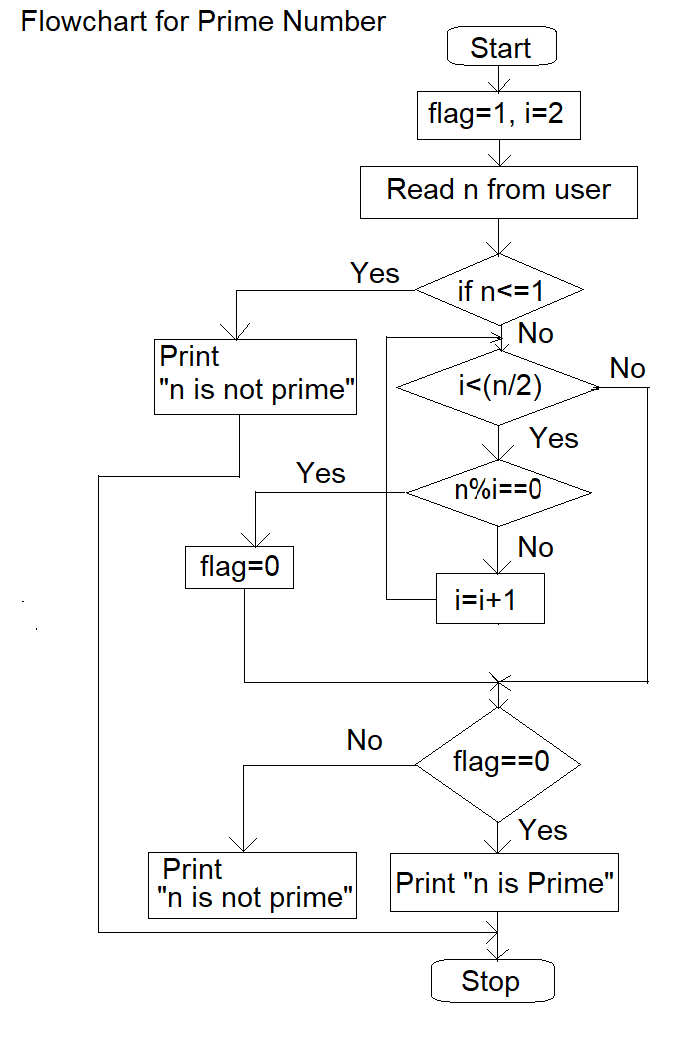
ENDIF



3. Program to check if the input number is prime or not.

Prime numbers are the numbers which are not divisible by another number except 1 and the number itself. Some of the prime numbers are 2,3,5,7,11,13,17 etc and an additional information 2 is the only even prime number.





**Algorithm**

Step 1: Start

Step 2: Initialize variables num,flag=1, j=2

Step 3: Read num from user

Step 4: If num<=1 // Any number less than 1 is not a prime number

Display "num is not a prime number"

Goto step 7

Step 5: Repeat the steps until j<[(n/2)+1]

5.1 If remainder of number divide j equals to 0,

Set flag=0

Goto step 6

5.2 j=j+1

Step 6: If flag==0,

Display num+" is not prime number"

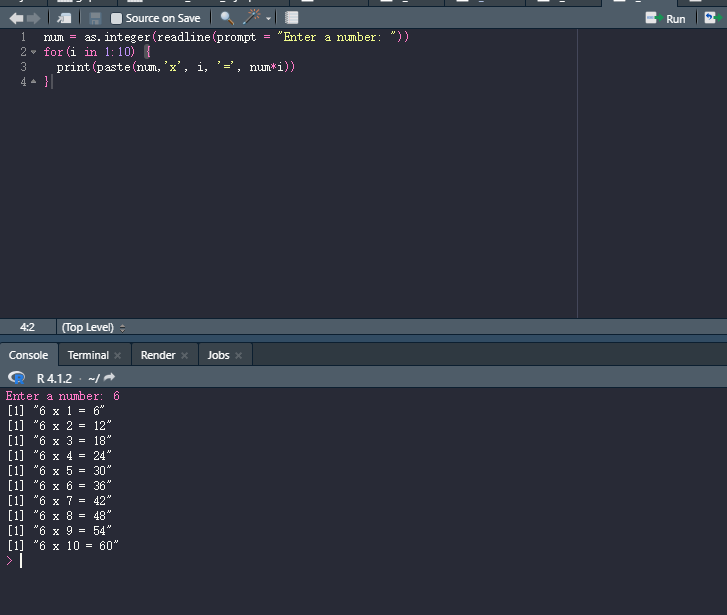
Else

Display num+" n is prime number"

Step 7: Stop

4. Program to find the multiplication table for a given number.

Self



#### Hints:

#### readline() function

We can read the input given by the user in the terminal with the readline() function.

Code:

input\_read <- readline()

#### 45 3 45

#### input\_read

#### ???

#### scan() function

We can also use the scan() function to read user input. This function, however, can only read numeric values and returns a numeric vector. If a non-numeric input is given, the function gives an error.

Code:

input\_scan <- scan()

User Input:

1: 34 54 65 75 23  
6:  
Read 5 items

#### print() functions

#### We can use the print() function to display the output to the terminal. The print() function is a generic function. 45 3 45

print(input\_read)

45 3 45

#### cat() function

We can also use the cat() function to display a string. The cat() function concatenates all of the arguments and forms a single string which it then prints. For example:

Code:

cat("hello", "this","is","RPD",12345,TRUE)

hello this is RPD 12345 TRUE

| Solution: |
| --- |

\*\*\*\*\*\*\*