**Practical – 1**

1. **Write a program to display student information.**

Code:

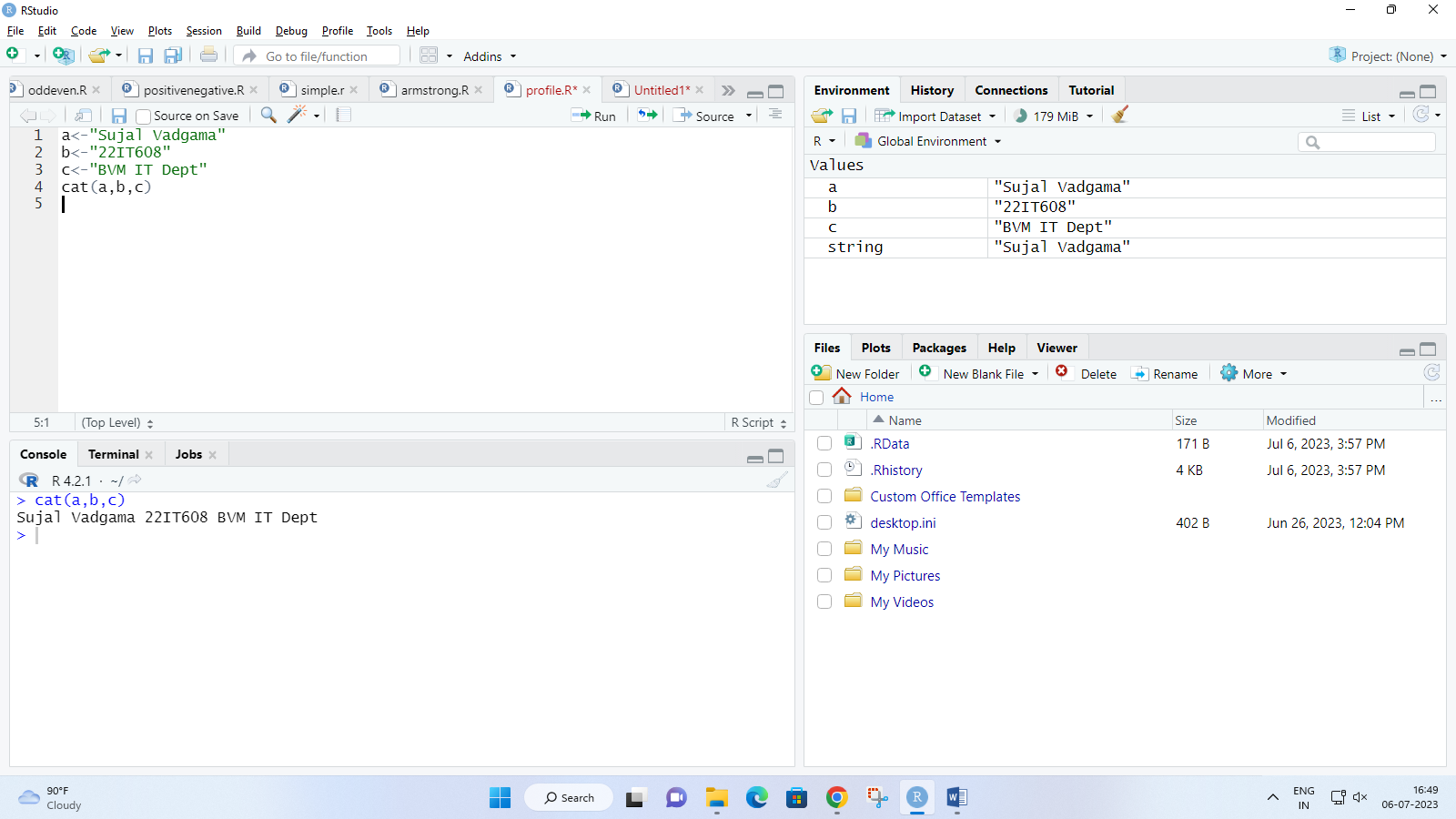
a<-"Sujal Vadgama"

b<-"22IT608"

c<-"BVM IT Dept"

cat(a,b,c)

Output :



1. **Write a program to find weather a number is even or odd.**

Code:

n <- as.integer(readline(prompt="Enter any number : "))

if(n%%2==0){

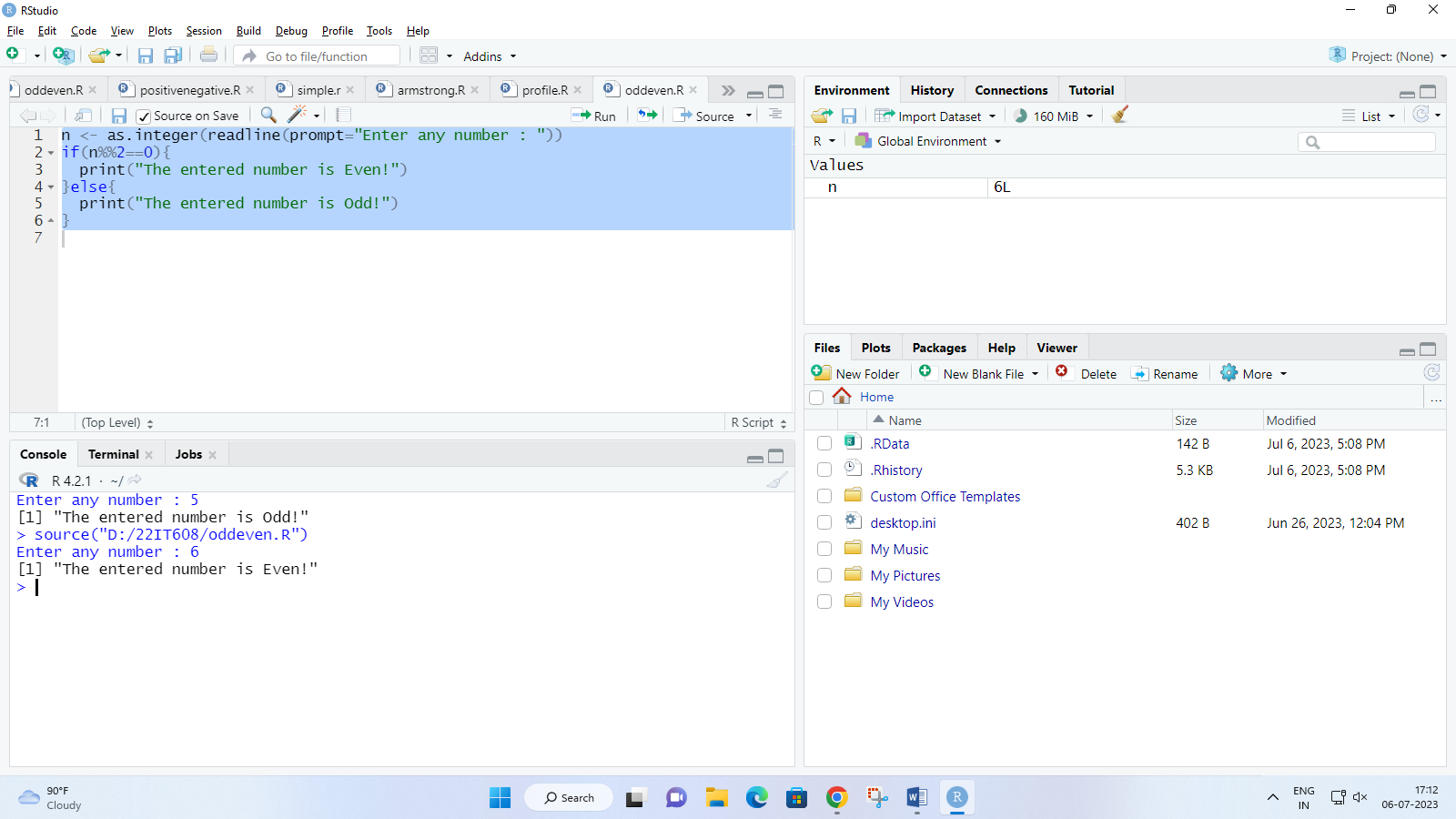
print("The entered number is Even!")

}else{

print("The entered number is Odd!")

}

Output:



1. **Write a program to create a calculator.**

Code :

a <- as.integer(readline(prompt="Enter 1st Number : "))

b <- as.integer(readline(prompt="Enter 2nd Number : "))

choice <- as.integer(readline(prompt="Enter Choice :\n 1 = Addition\n 2 = Subtraction\n 3 = Multiplication\n 4 = Division\n 5 = Modulo\n "))

switch(choice,

"1"=cat("Addition",a+b),

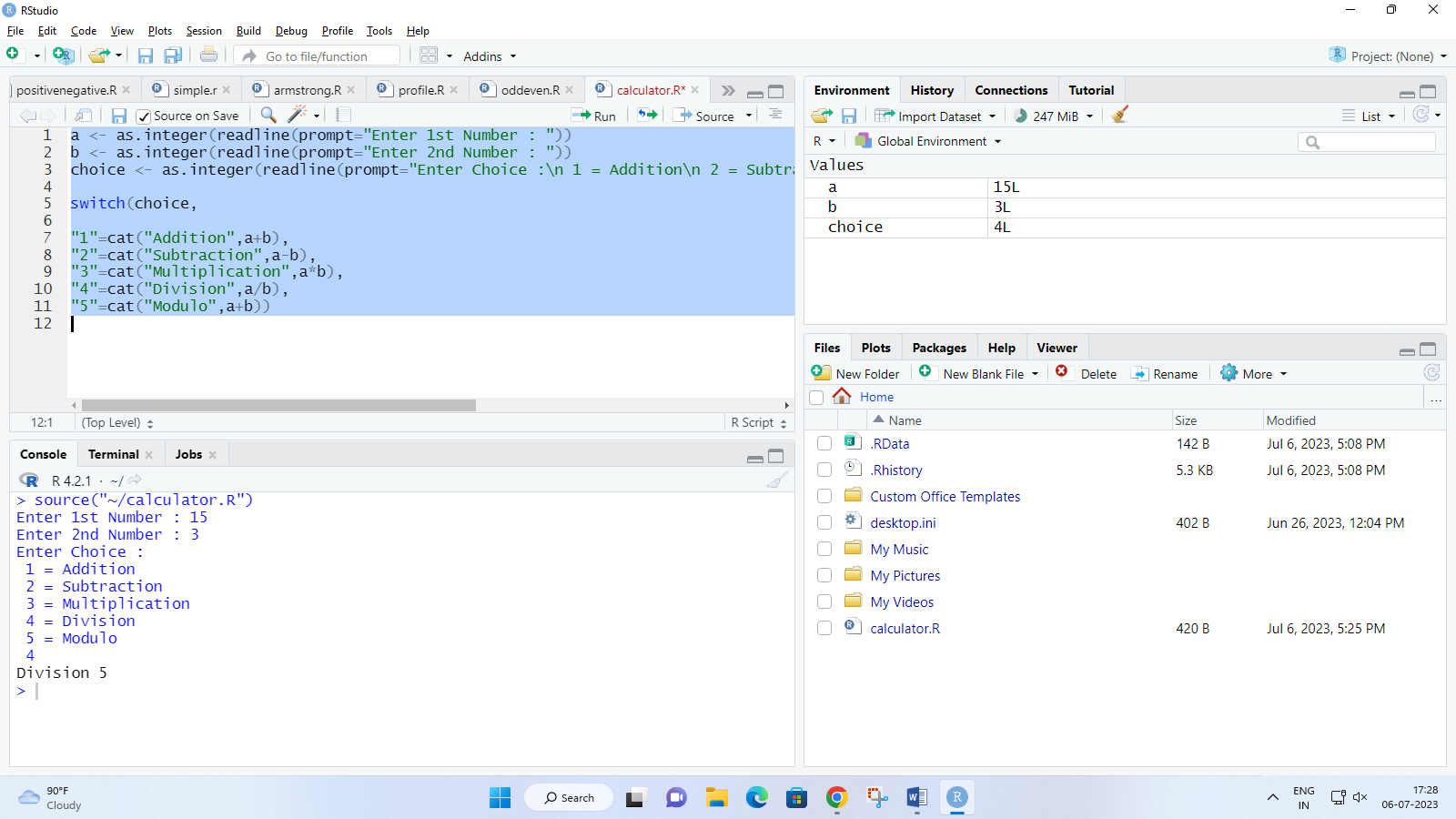
"2"=cat("Subtraction",a-b),

"3"=cat("Multiplication",a\*b),

"4"=cat("Division",a/b),

"5"=cat("Modulo",a+b))

Output :



1. Write a program to find greatest of 2 number.

Code :

a <- as.integer(readline(prompt="Enter 1st Number ; "))

b <- as.integer(readline(prompt="Enter 2nd Number ; "))

if(a<b){

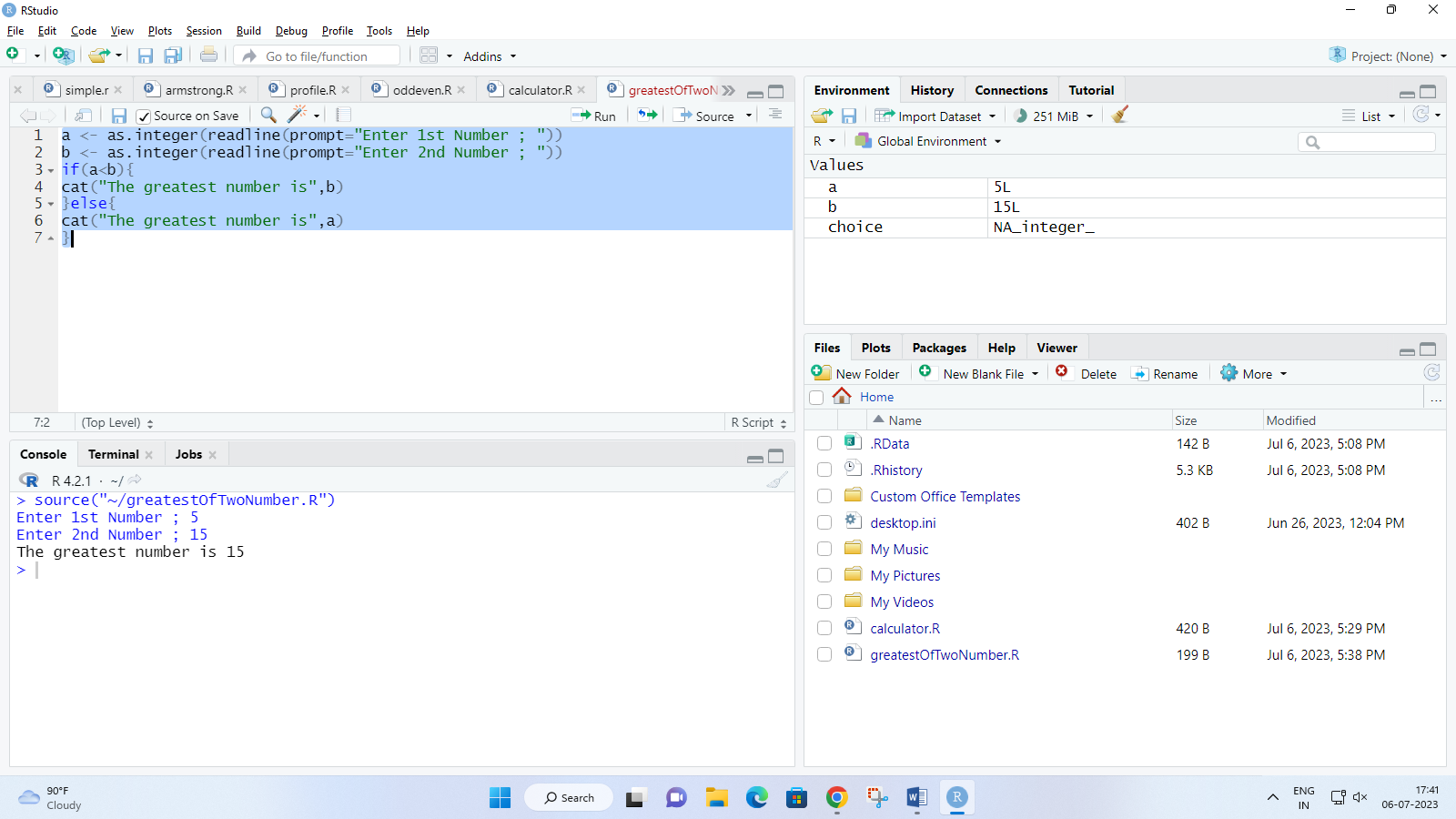
cat("The greatest number is",b)

}else{

cat("The greatest number is",a)

}

Output :



1. Write a program to calculate grade of student with help of marks scored.

Code :

eng <- as.integer(readline(prompt="Enter Marks of English : "))

mat <- as.integer(readline(prompt="Enter Marks of Maths : "))

sci <- as.integer(readline(prompt="Enter Marks of Science : "))

hin <- as.integer(readline(prompt="Enter Marks of Hindi : "))

guj <- as.integer(readline(prompt="Enter Marks of Gujarati : "))

if(eng >100 | mat > 100 | sci > 100 | hin > 100 | guj > 100){

cat("Please enter marks correctly!")

}else{

if(eng > 32){

cat("Please clear English!!!\nGrade : FF")

}else if(mat < 32){

cat("Please clear Maths!!!\nGrade : FF")

}else if(sci < 32){

cat("Please clear Science!!!\nGrade : FF")

}else if(hin < 32){

cat("Please clear Hindi!!!\nGrade : FF")

}else if(guj < 32){

cat("Please clear Gujarati!!!\nGrade : FF")

}else{

total <- eng + mat + sci + hin + guj

avg <- total/5

cat("Total marks scored : ",total,"\n",

"Percentage : ",avg,"%\n")

if(avg<=100 & avg>90){

cat("Grade : AA\n Success! Cleared examination with 1st class Distiction.")

}else if(avg<=90 & avg>=80){

cat("Grade : AB")

}else if(avg<=80 & avg>=70){

cat("Grade : BB")

}else if(avg<=70 & avg>=60){

cat("Grade : BC")

}else if(avg<=60 & avg>=50){

cat("Grade : CC")

}else if(avg<=50 & avg>=40){

cat("Grade : CD")

}else if(avg<=40 & avg>=35){

cat("Grade : DD")

}else if(avg<35){

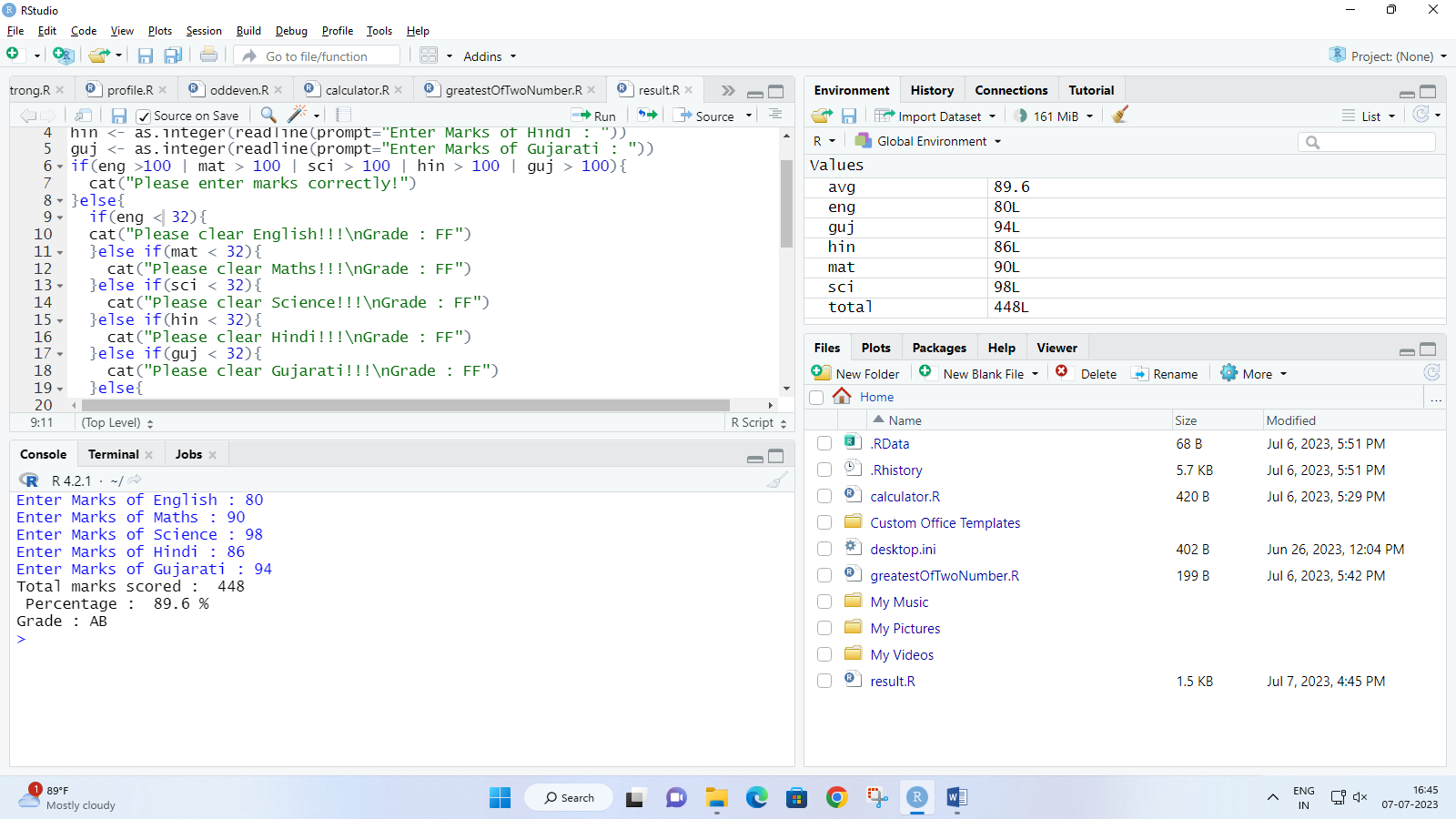
cat("Grade : FF\n Sorry! Did not clear examination.")

}

}

}

Output :



1. Write a program to print Fibonacci series of a entered number

Code :

n <- as.integer(readline(prompt="Enter a Number to print it's fibonacci series : "))

a <- 0

b <- 1

cat(a,b,"")

while(n != 0){

c = a + b

a = b

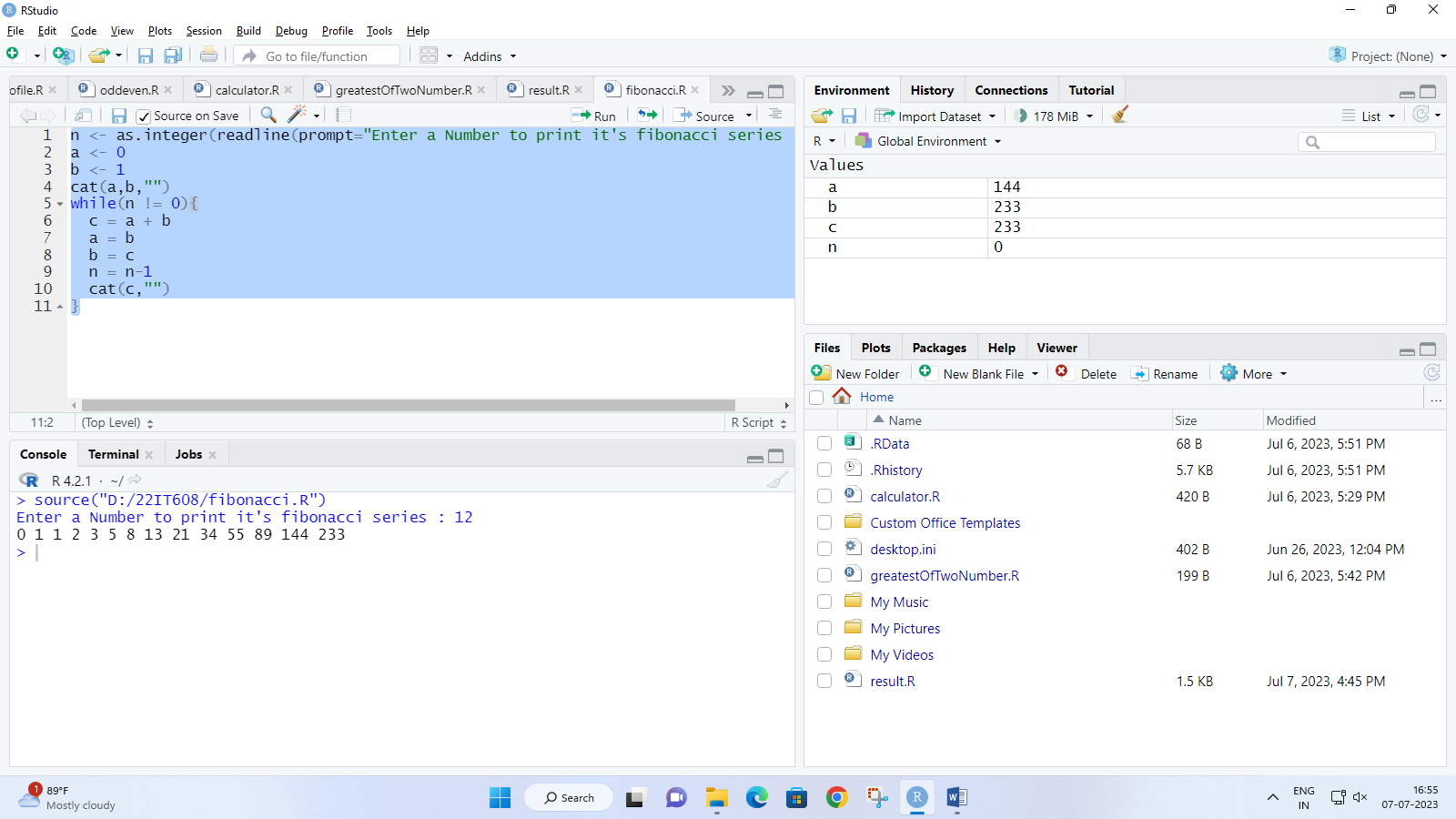
b = c

n = n-1

cat(c,"")

}

Output :



1. Write a program to check weather entred number is palindrome or not.

Code :

n <- as.integer(readline(prompt="Enter a Number : "))

num <- n

rev <- 0L

while(n != 0){

rem = n %% 10

rev = rem + (rev\*10)

n = as.integer(n/10)

}

if(num == rev){

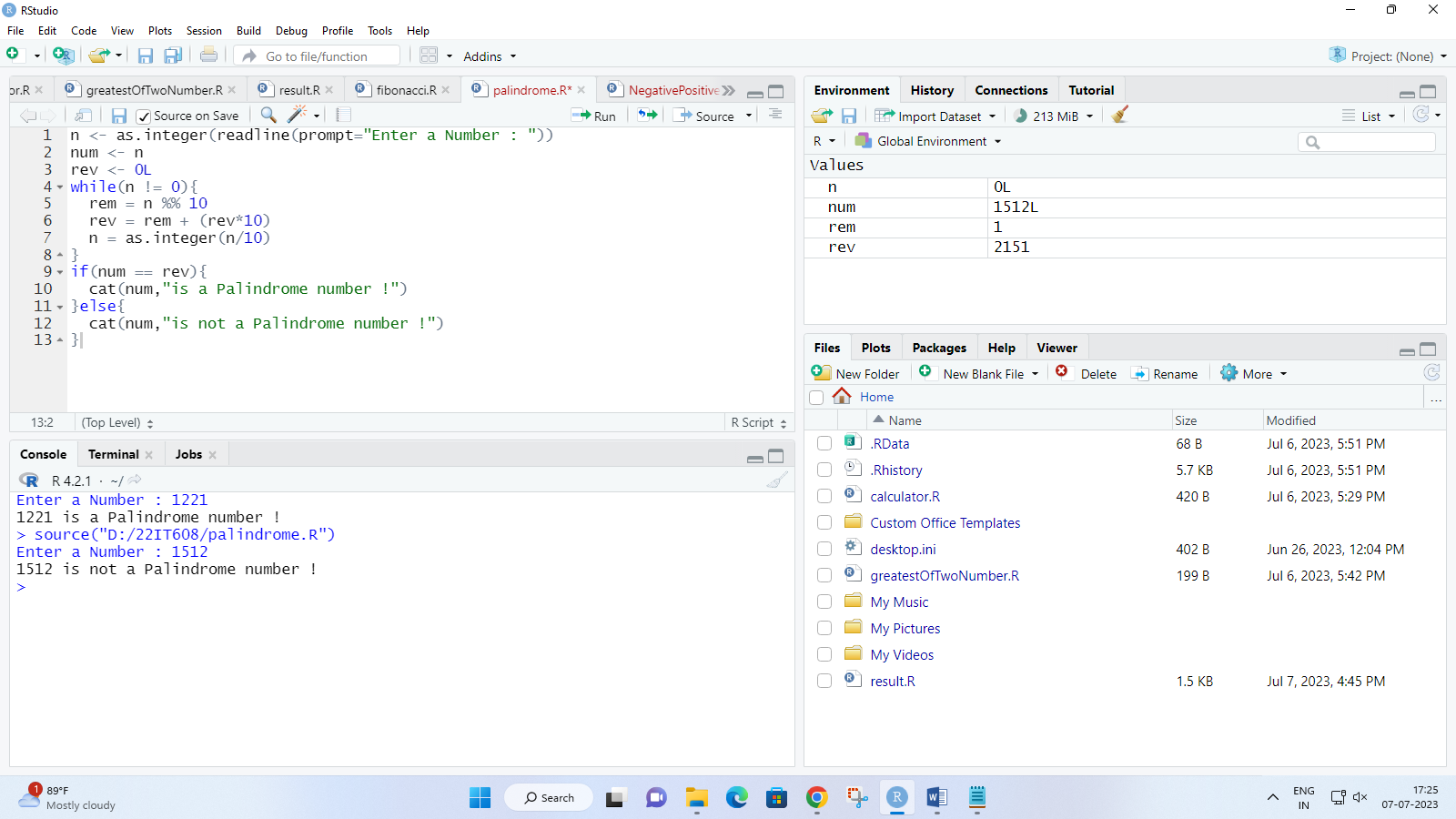
cat(num,"is a Palindrome number !")

}else{

cat(num,"is not a Palindrome number !")

}

Output :



1. Write a program to check weather a entered number is Armstrong or not.

Code :

n <- as.integer(readline(prompt="Enter a Number : "))

org <- n

temp <- n

len <- 0

sum <- 0L

while(temp != 0){

temp <- floor(temp/10)

len <- len+1

}

while(n != 0){

remainder <- (n%%10)

sum <- (sum + (remainder^len))

n <- as.integer(n/10)

}

if(org == sum){

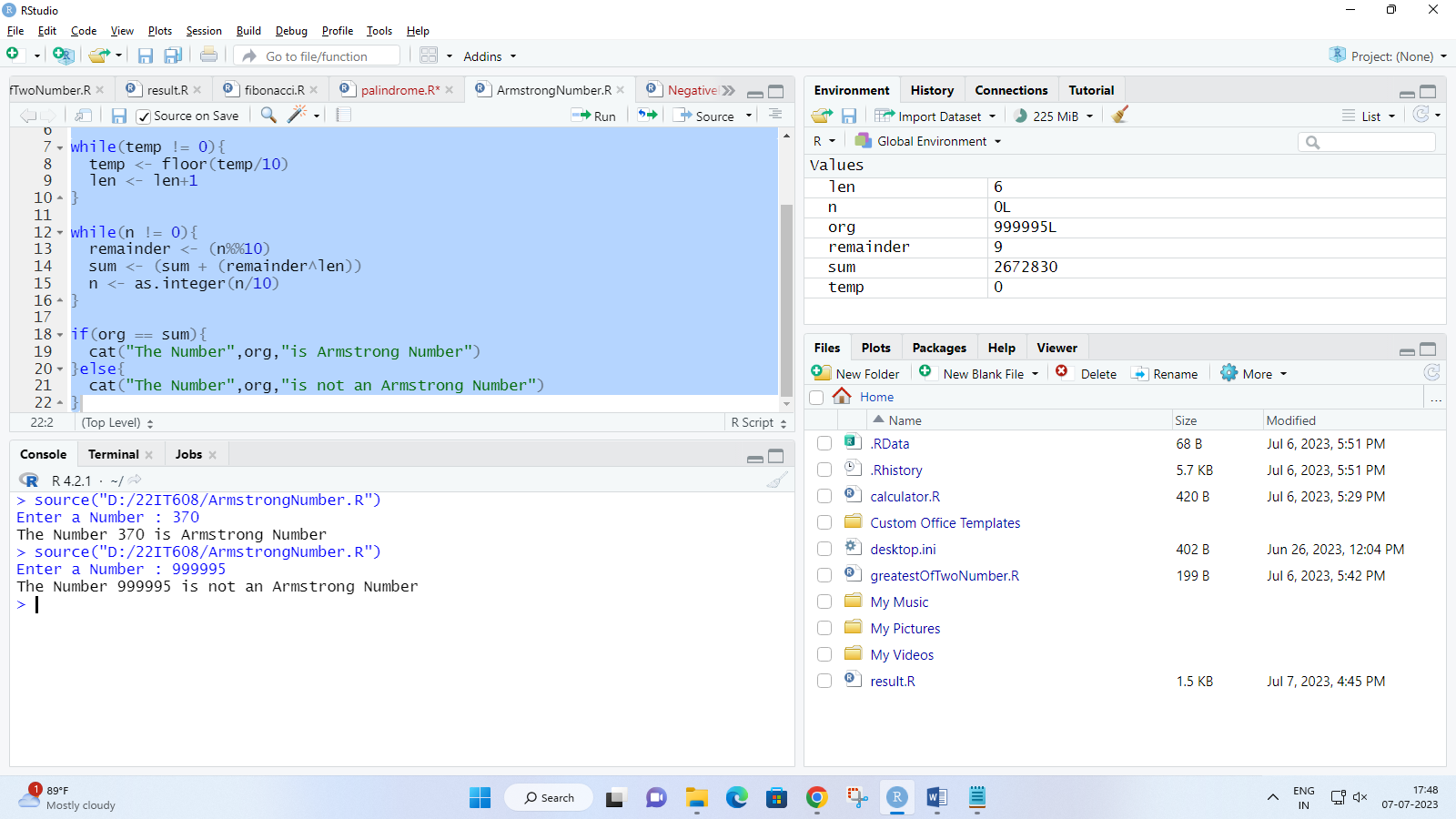
cat("The Number",org,"is Armstrong Number")

}else{

cat("The Number",org,"is not an Armstrong Number")

}

Output :



1. Write a program to find factorial of a number (recursion)

Code :

fact <- function(n) {

if(n==0 || n==1)

{

return(1)

} else{

return(n\*fact(n-1))

}

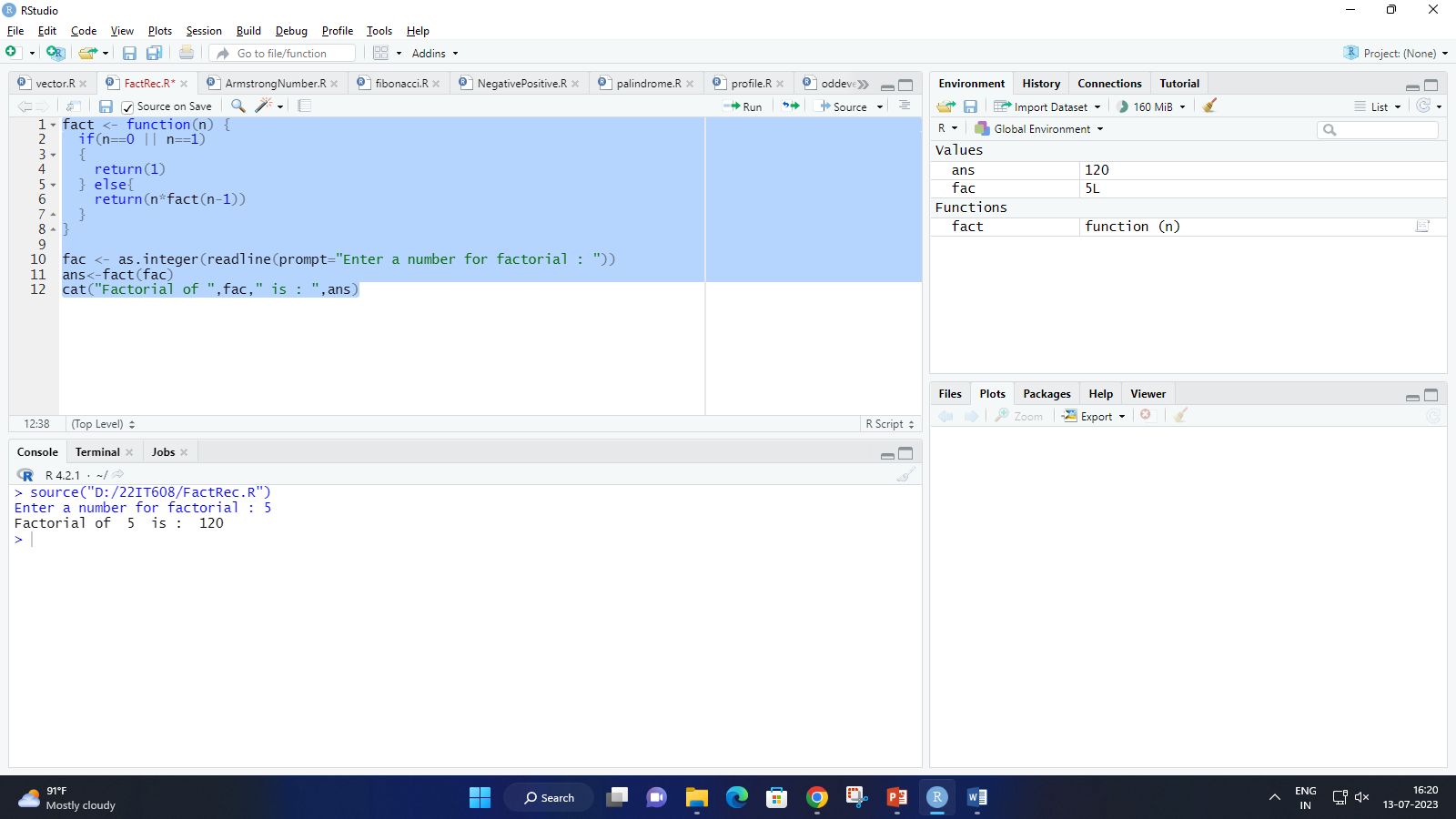
}

fac <- as.integer(readline(prompt="Enter a number for factorial : "))

ans<-fact(fac)

cat("Factorial of ",fac," is : ",ans)

Output :



1. Create two vectors of different lengths.

Code :

vec1 <- c(1,2,3)

vec2 <- c(10,20,30,40,50,60,70,80,90)

cat("1st Vector\n",vec1)

cat("\n\n2nd Vector\n",vec2)

Output:



1. Create a program that takes above two vectors as input to the array.

Code :

vec1 <- c(1,2,3)

vec2 <- c(10,20,30,40,50,60,70,80,90)

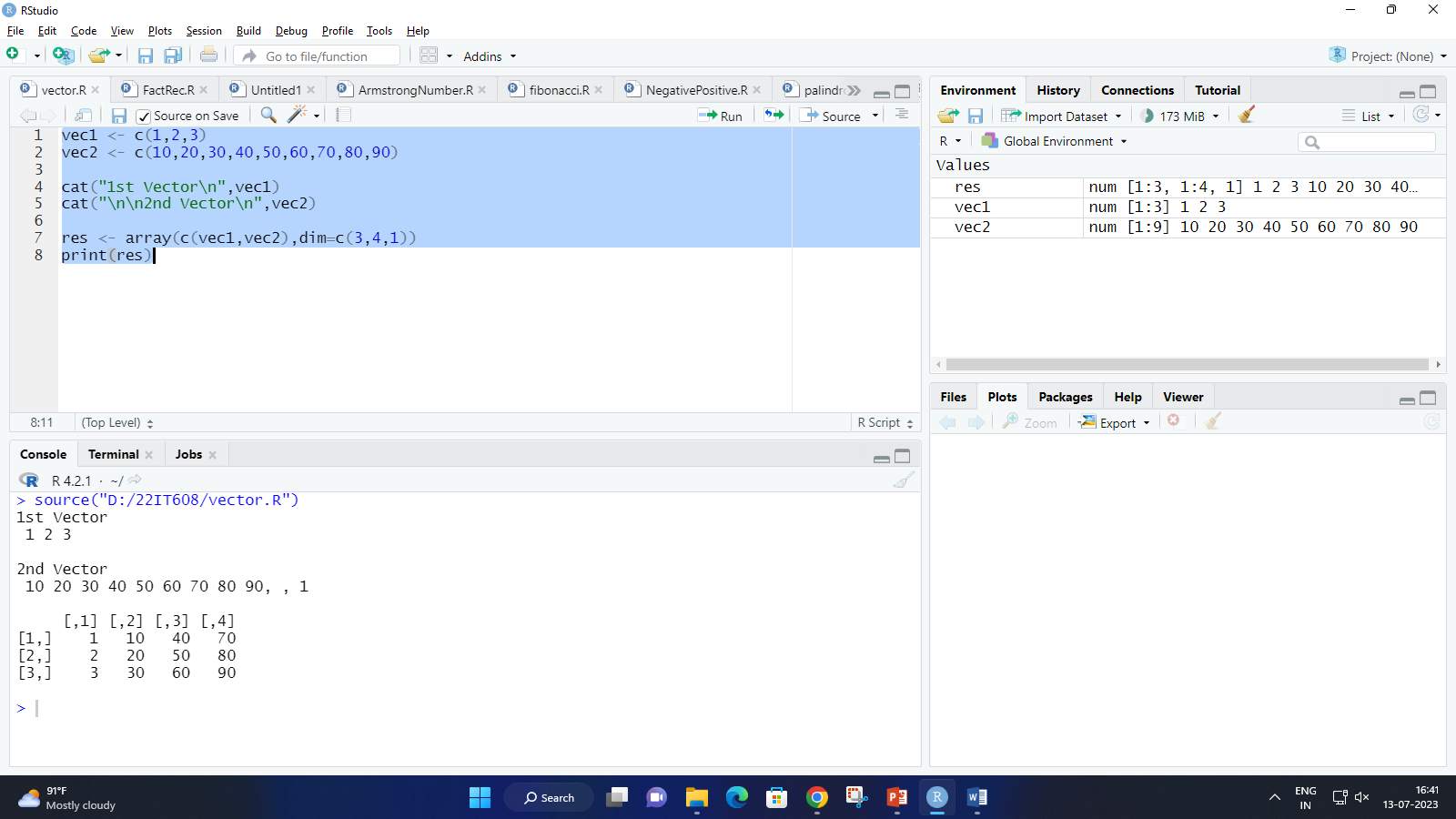
cat("1st Vector\n",vec1)

cat("\n\n2nd Vector\n",vec2)

res <- array(c(vec1,vec2),dim=c(3,4,1))

print(res)

Output:



1. Write a program that creates a list containing a vector, a matrix and a list.

Code:

list\_data<-list(c("Darshak","Roshan","Darshan"),

matrix(c(40,80,60,70,90,80),nrow=3),

c("BCA","MCA","MBA"))

list\_data2<-list(c("Jaivik","Ali","Rahil"),

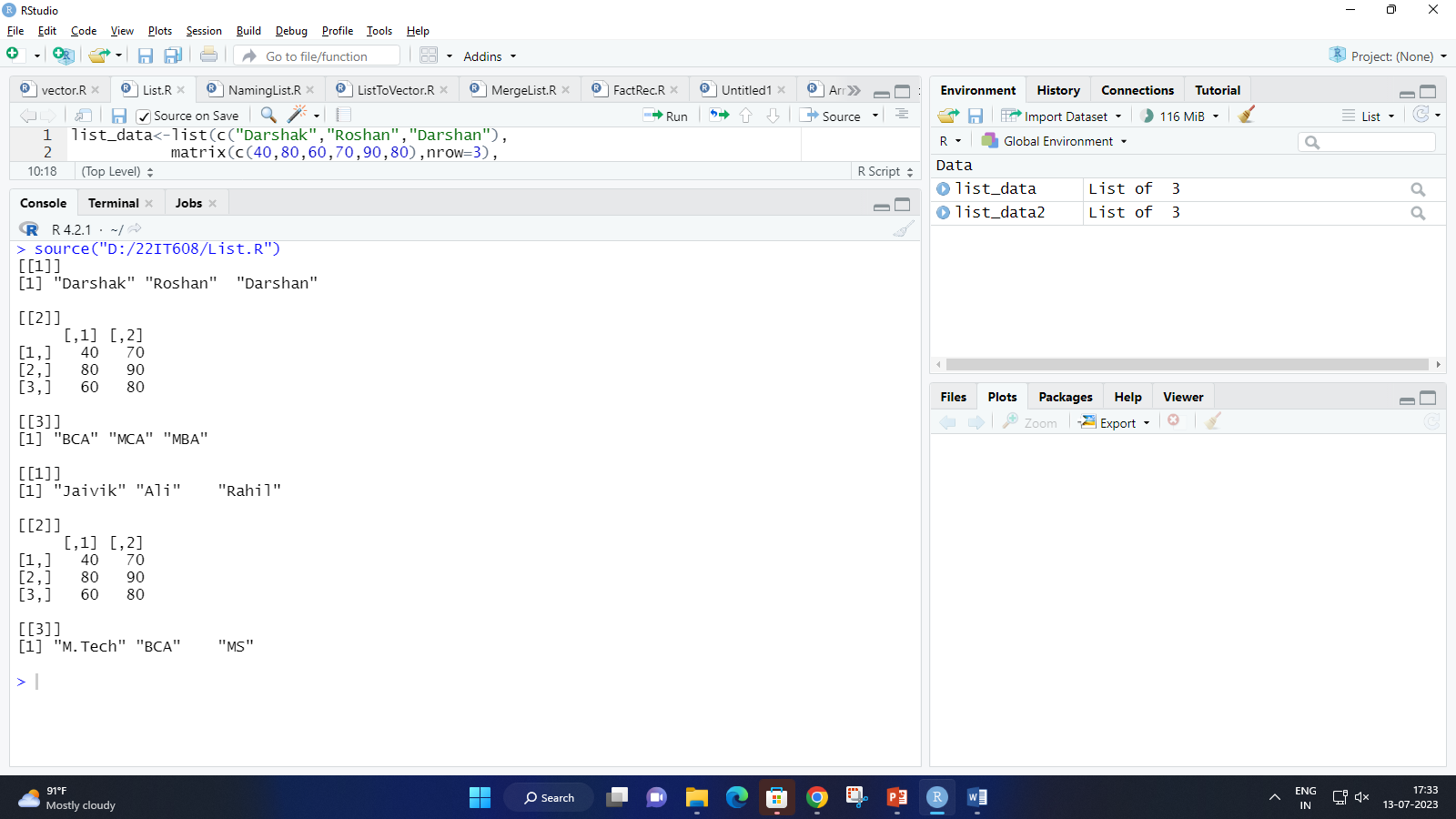
matrix(c(40,80,60,70,90,80),nrow=3),

c("M.Tech","BCA","MS"))

print(list\_data)

print(list\_data2)

Output:



1. Write a program to name a list and access the first index of list.

Code:

list\_data<-list(c("Darshak","Roshan","Darshan"),

matrix(c(40,80,60,70,90,80),nrow=3),

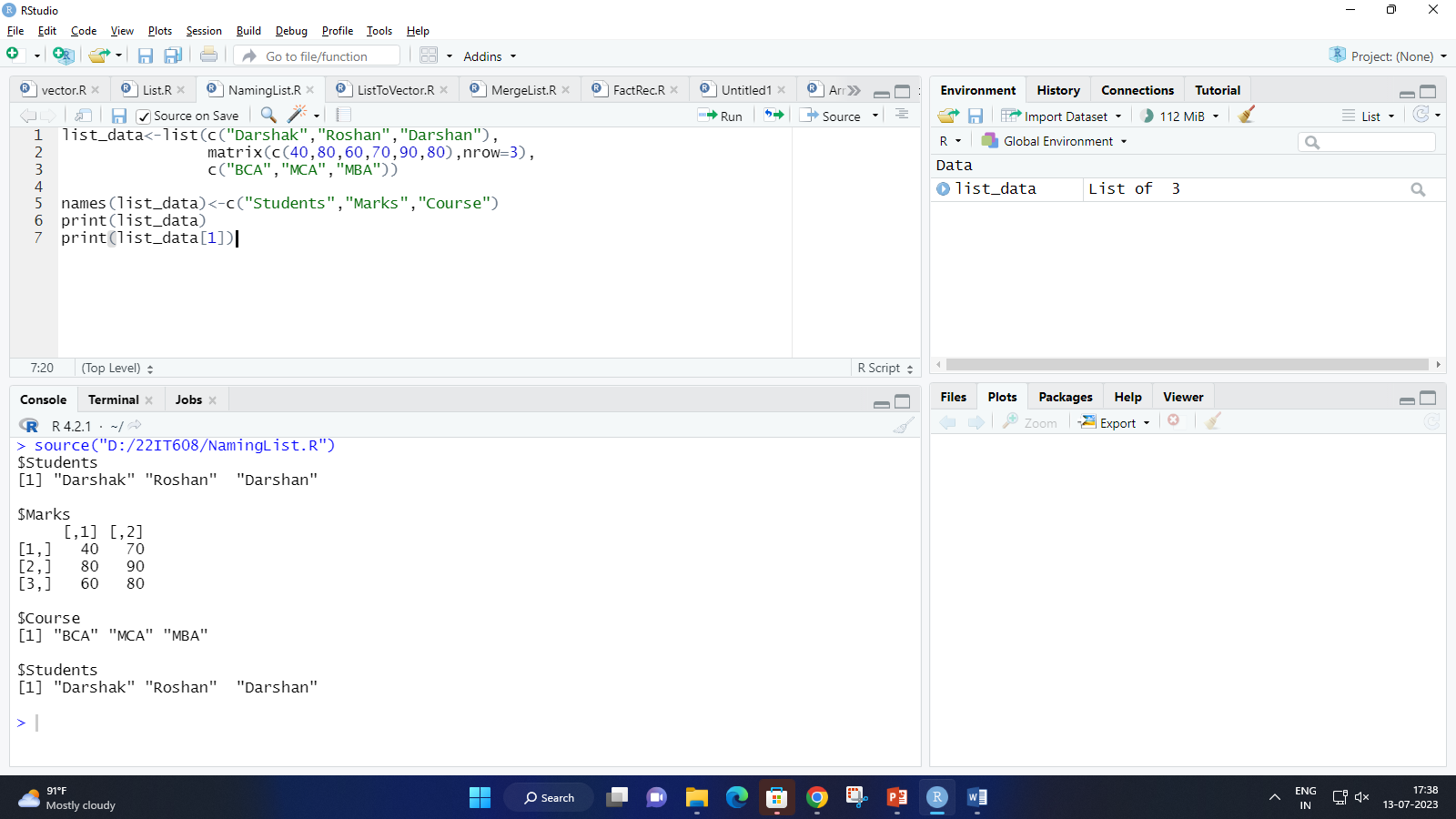
c("BCA","MCA","MBA"))

names(list\_data)<-c("Students","Marks","Course")

print(list\_data)

print(list\_data[1])

Output:



1. Write a program to convert a List to Vector.

Code:

list\_data<-list(c("Darshak","Roshan","Darshan"),

matrix(c(40,80,60,70,90,80),nrow=3),

c("BCA","MCA","MBA"))

list\_data2<-list(c("Jaivik","Ali","Rahil"),

matrix(c(40,80,60,70,90,80),nrow=3),

c("M.Tech","BCA","MS"))

v1 <- unlist(list\_data)

v2 <- unlist(list\_data2)

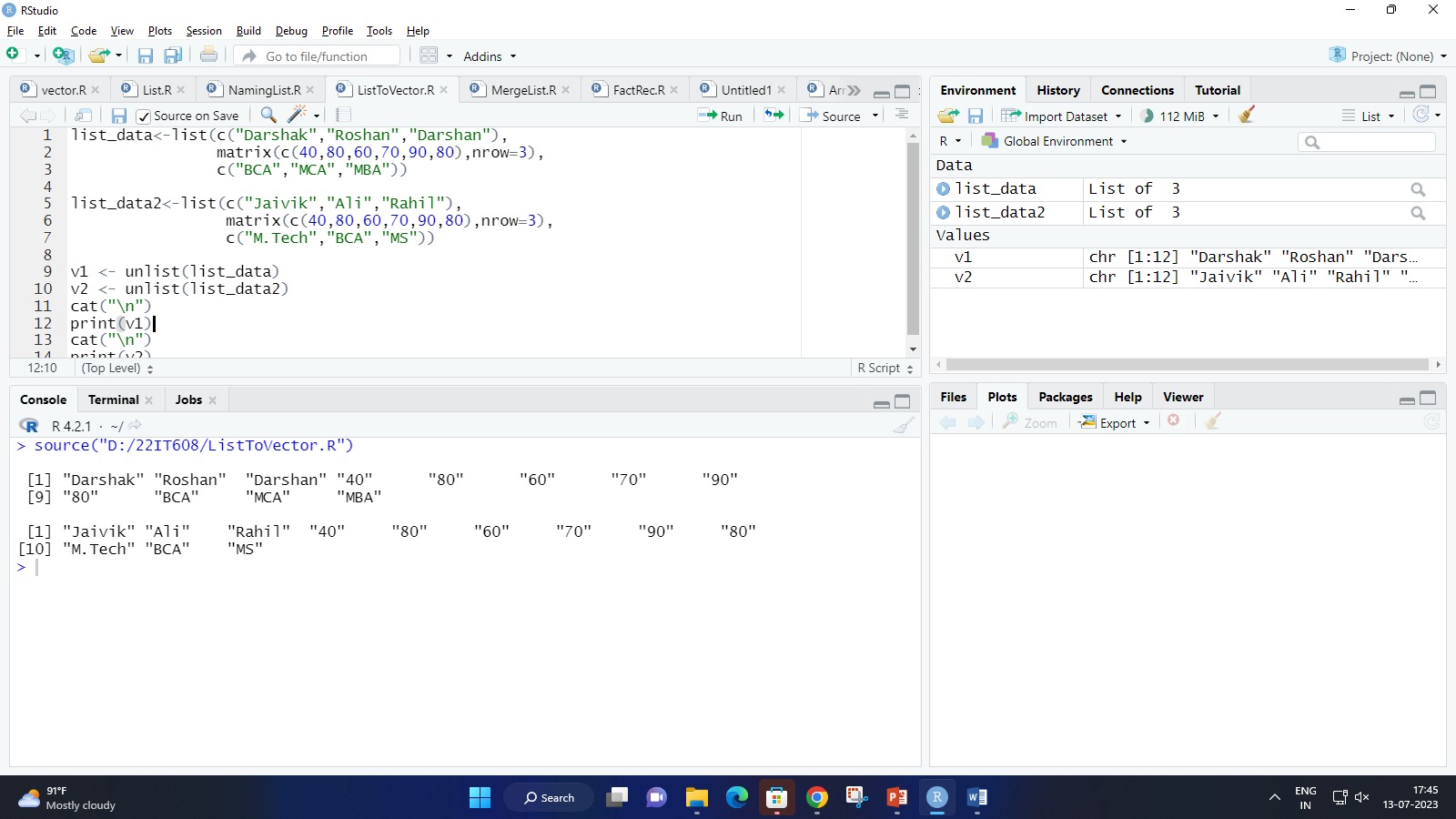
cat("\n")

print(v1)

cat("\n")

print(v2)

Output:



1. Write a program to Merge List.

Code:

list\_data<-list(c("Darshak","Roshan","Darshan"),

matrix(c(40,80,60,70,90,80),nrow=3),

c("BCA","MCA","MBA"))

list\_data2<-list(c("Jaivik","Ali","Rahil"),

matrix(c(40,80,60,70,90,80),nrow=3),

c("M.Tech","BCA","MS"))

cat("\nMearge List\n")

A<-list(list\_data,list\_data2)

cat("\n\n")

print(A)

Output:



1. Write a program that arranges matrix data sequentially by row.

Code :

a <- matrix(c(11,12,13,14,15,16),2,3,TRUE)

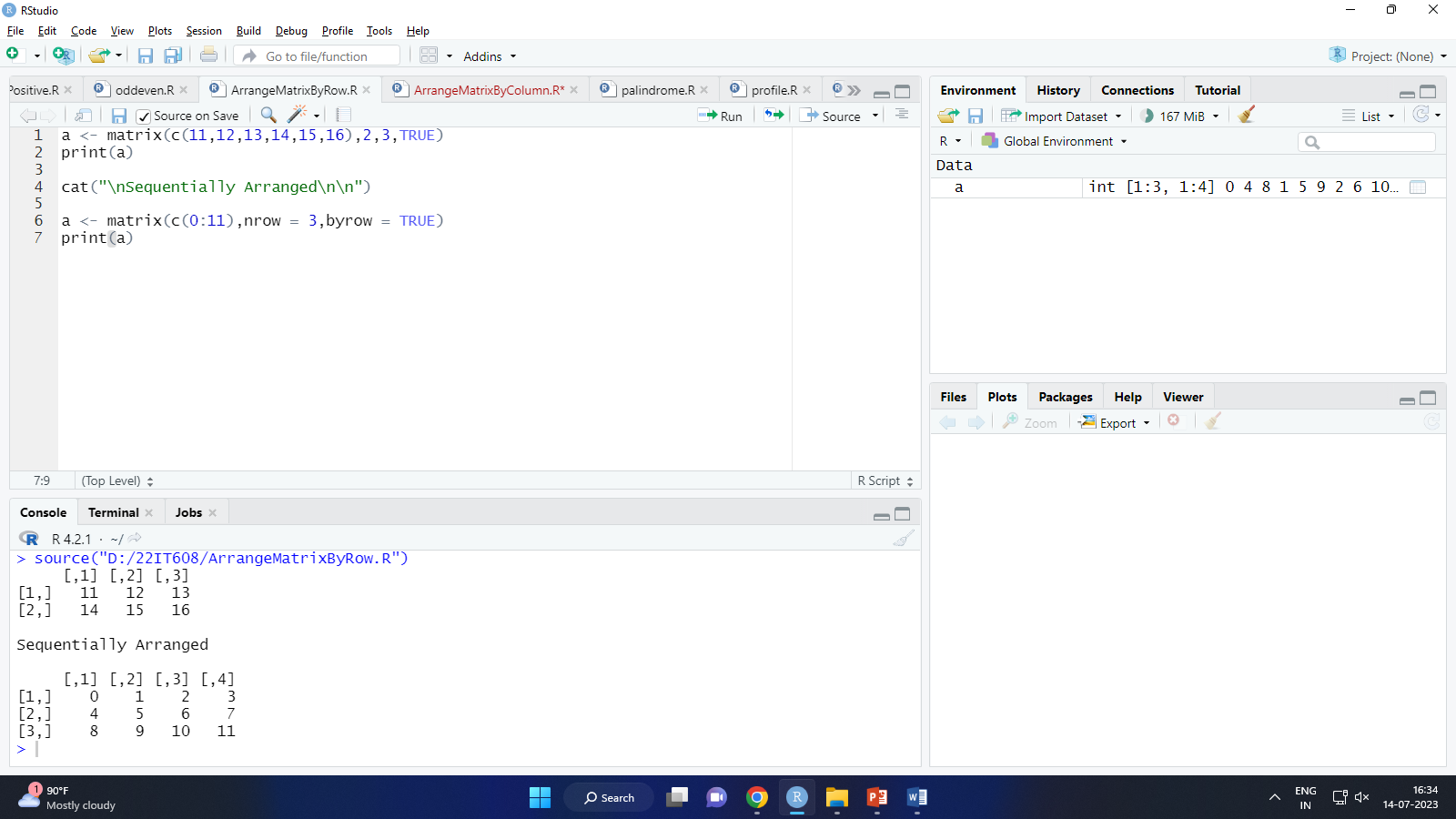
print(a)

cat("\nSequentially Arranged\n\n")

a <- matrix(c(0:11),nrow = 3,byrow = TRUE)

print(a)

Output :



1. Write a program that arranges matrix data sequentially by column.

Code:

a <- matrix(c(11,12,13,14,15,16),2,3,TRUE)

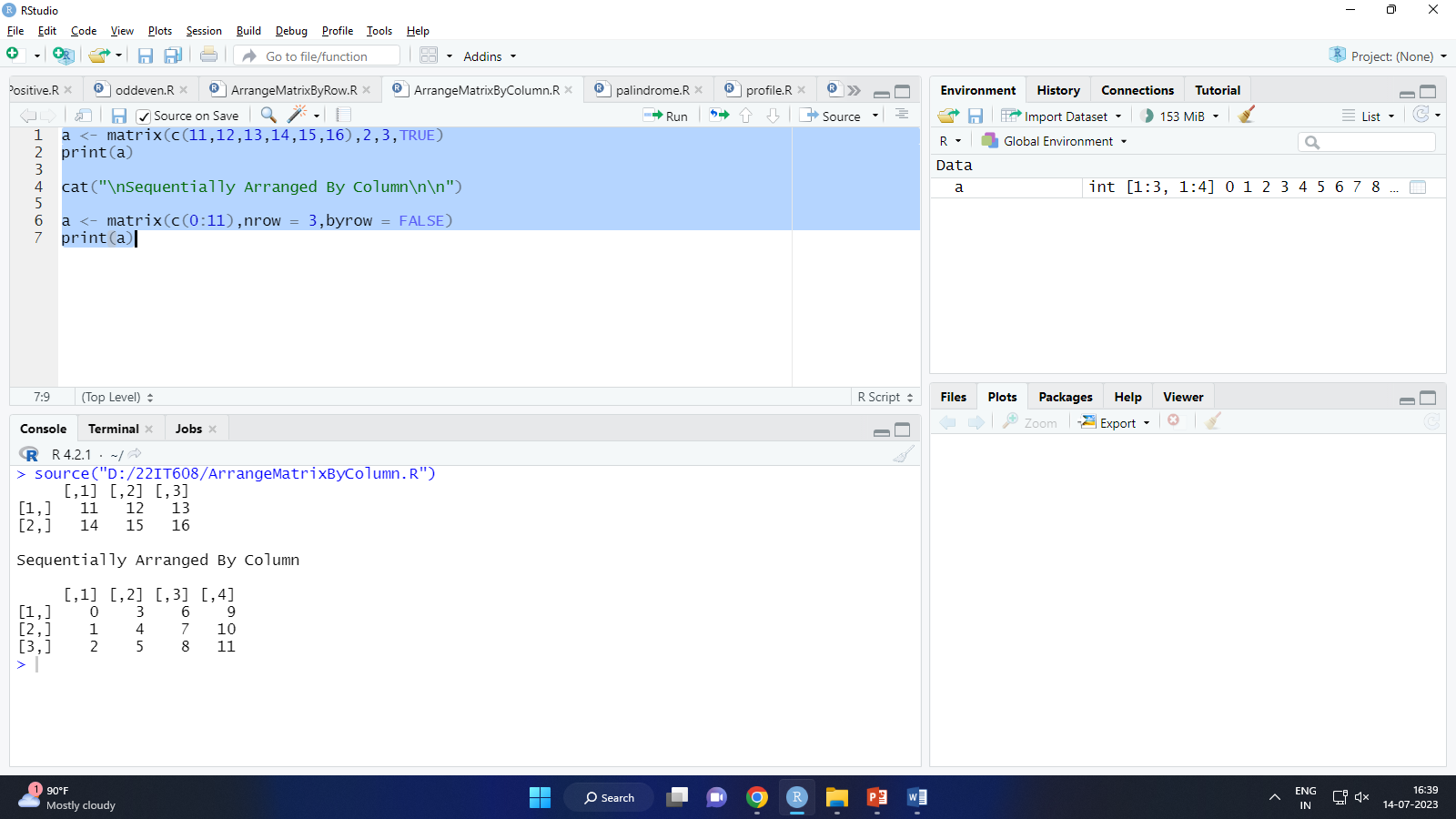
print(a)

cat("\nSequentially Arranged By Column\n\n")

a <- matrix(c(0:11),nrow = 3,byrow = FALSE)

print(a)

Output :



1. Write a program to name rows and columns.

Code:

a <- matrix(c(11,12,13,14,15,16),2,3,TRUE)

print(a)

cat("\nSequentially Arranged By Column\n\n")

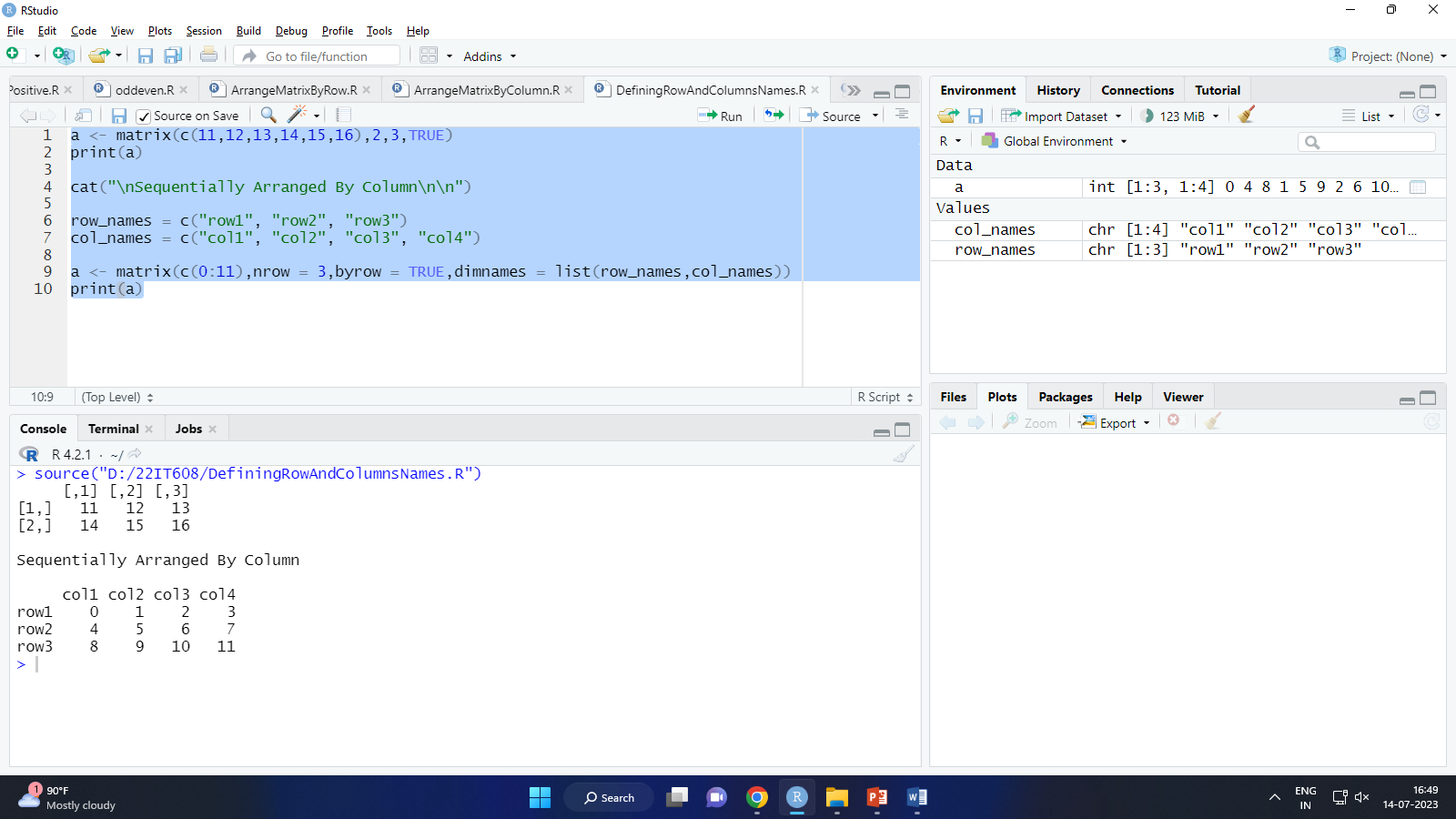
row\_names = c("row1", "row2", "row3")

col\_names = c("col1", "col2", "col3", "col4")

a <- matrix(c(0:11),nrow = 3,byrow = TRUE,dimnames = list(row\_names,col\_names))

print(a)

Output :



1. Write a program that accesses data by row and column.

Code:

a <- matrix(c(11,12,13,14,15,16),2,3,TRUE)

print(a)

cat("\nSequentially Arranged By Column\n\n")

row\_names = c("row1", "row2", "row3")

col\_names = c("col1", "col2", "col3", "col4")

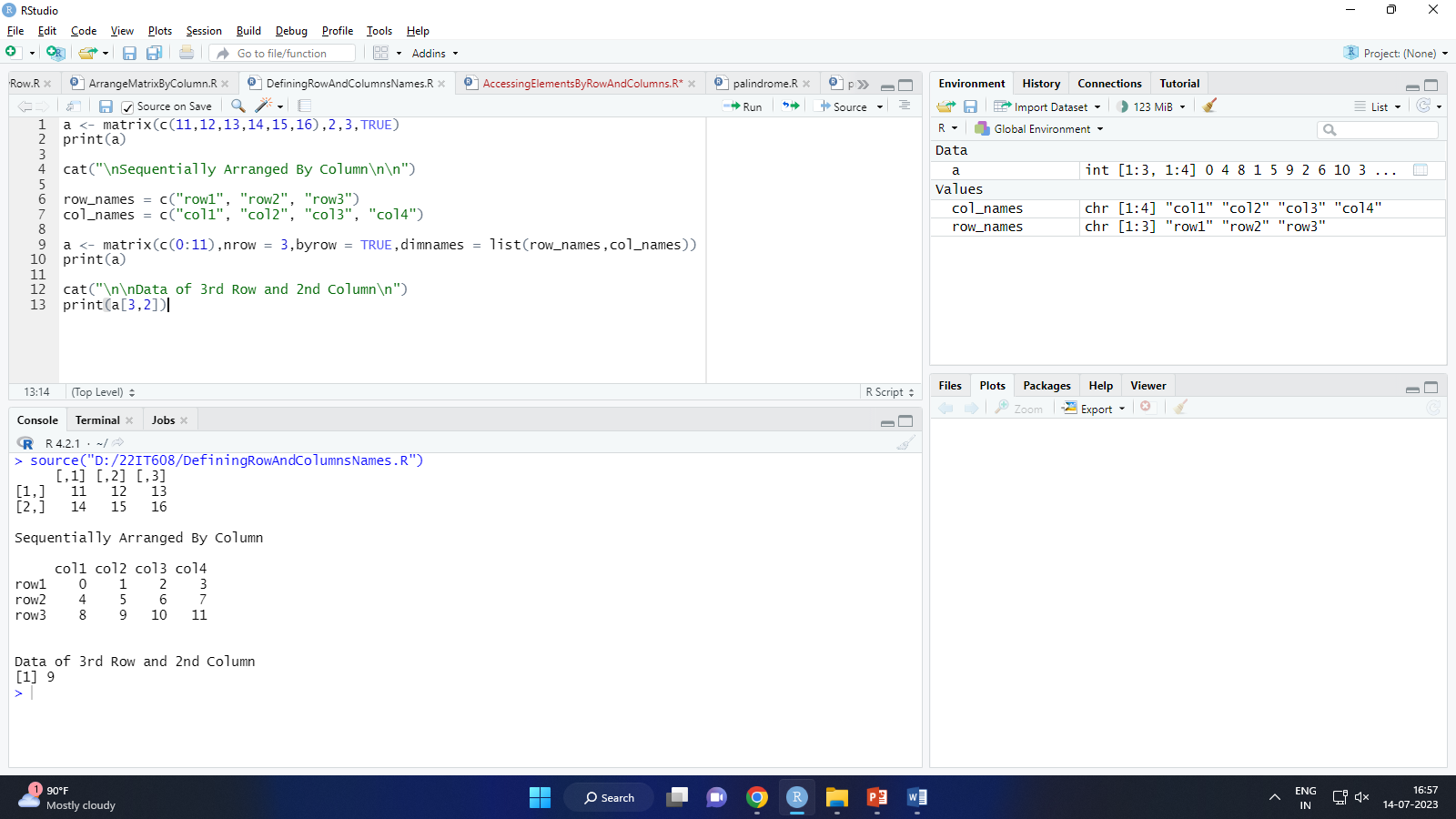
a <- matrix(c(0:11),nrow = 3,byrow = TRUE,dimnames = list(row\_names,col\_names))

print(a)

cat("\n\nData of 3rd Row and 2nd Column\n")

print(a[3,2])

Output :



1. Write a program that adds a column and a row and transposes a matrix.

Code:

a <- matrix(c(0:11),nrow = 3,byrow = TRUE,)

print(a)

cat("\n\nAdding a row in Matrix\n")

add\_row <- rbind(a,c(100,200,300,900))

print(add\_row)

cat("\n\nAdding a column in Matrix\n")

add\_col <- cbind(a,c(400,500,600))

print(add\_col)

cat("\n\nTranspose of a Matrix\n")

transpose<-t(a)

print(transpose)

Output :



1. Write a program that performs mathematical operations on matrix.

Code:

R <- matrix(c(5:16),nrow=4,ncol=3)

S <- matrix(c(1:12),nrow=4,ncol=3)

cat("\n\nAddition\n")

sum <- R+S

print(sum)

cat("\n\nSubtraction\n")

sub <- R-S

print(sub)

cat("\n\nMultiplication\n")

mul <- R\*S

print(mul)

cat("\n\nMultiplication by Constant\n")

mul1 <- R\*12

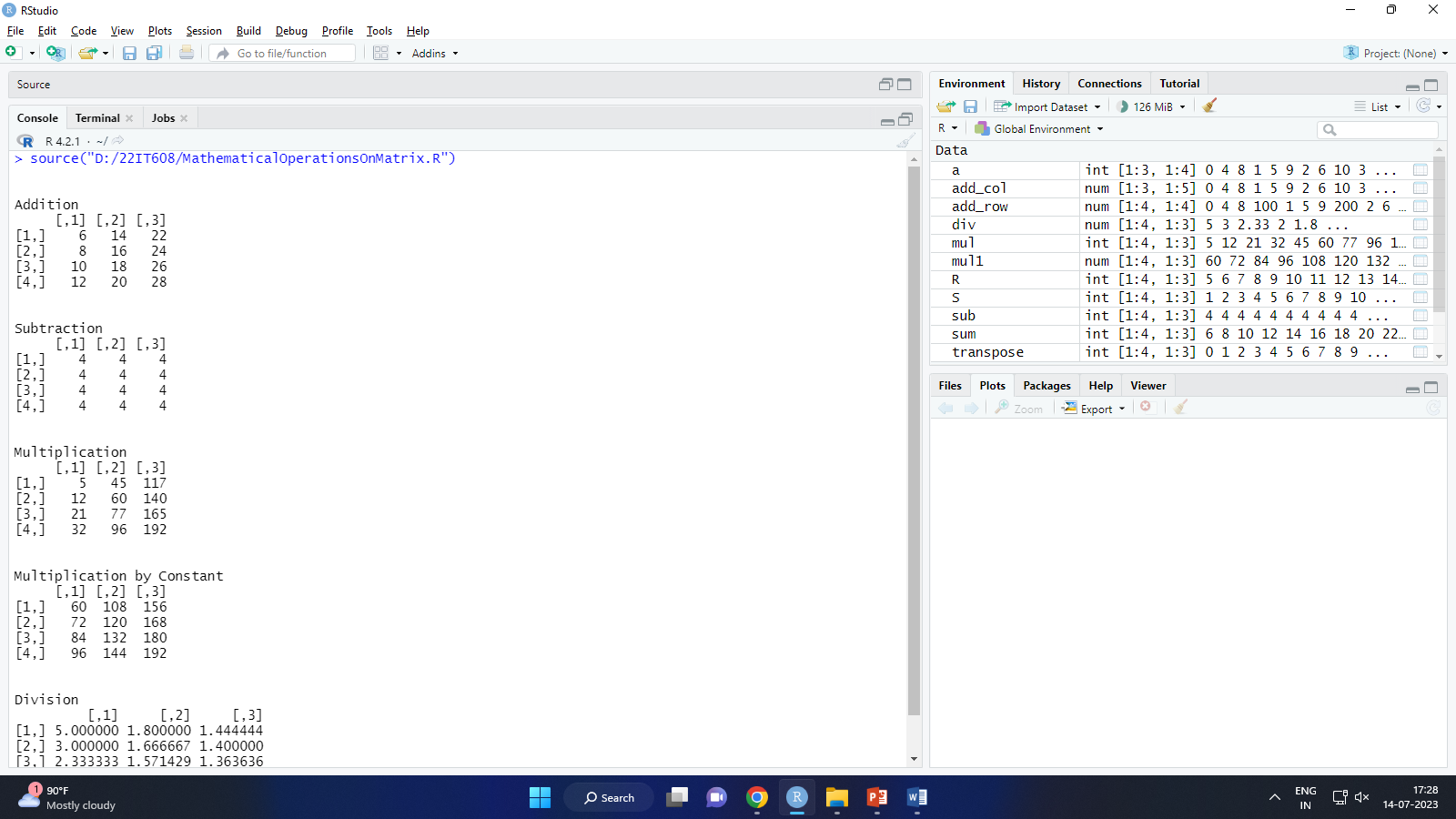
print(mul1)

cat("\n\nDivision\n")

div <- R/S

print(div)

Output :

1. Write a program to create a data frame.

Code:

std.data <- data.frame(

Id = c(1:5),

Name = c("Darshak","Roshan","Darshan","Jaivik","Sujal"),

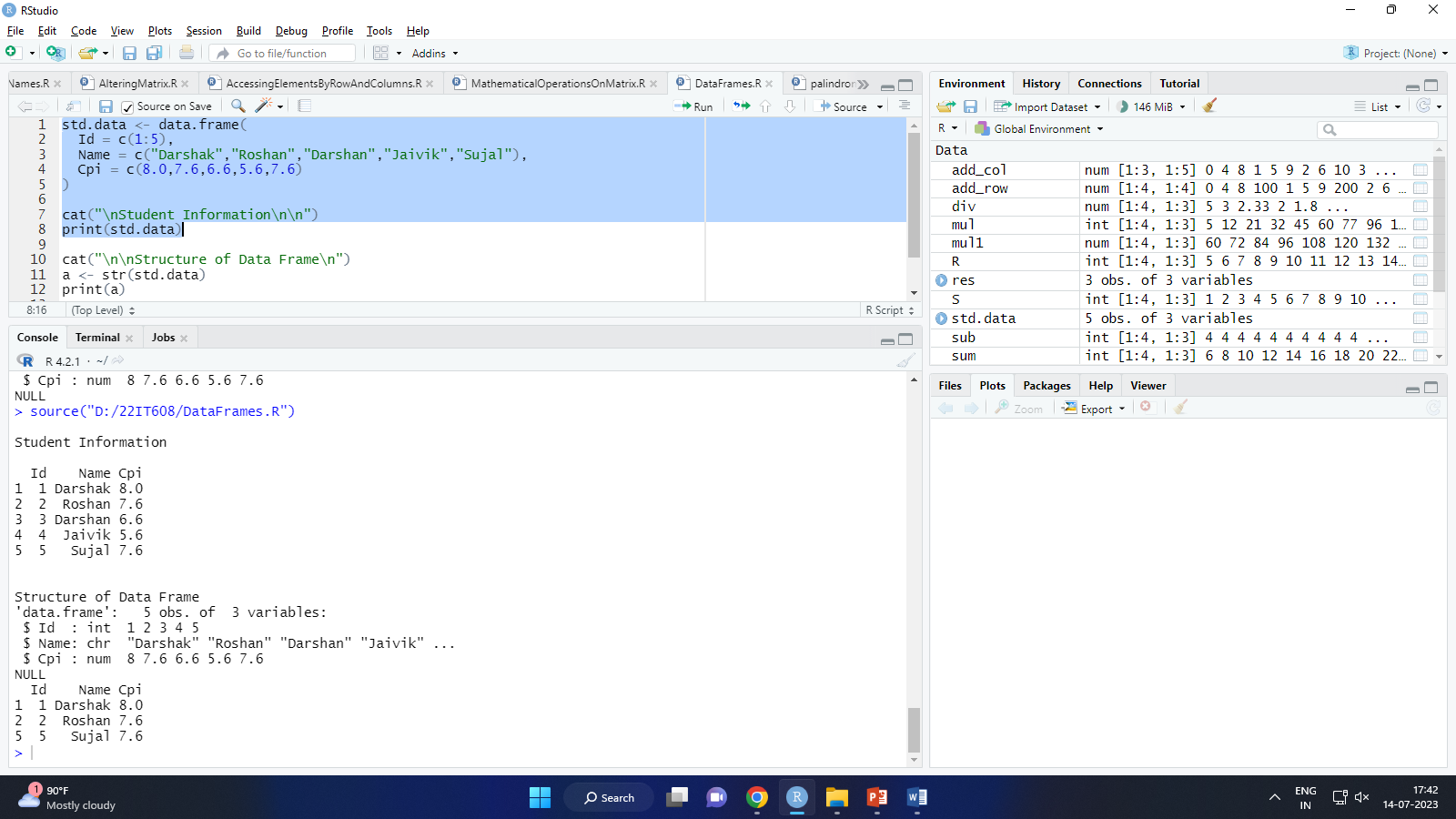
Cpi = c(8.0,7.6,6.6,5.6,7.6)

)

cat("\nStudent Information\n\n")

print(std.data)

Output :



1. Write a program to print the structure of dataframe.

Code:

std.data <- data.frame(

Id = c(1:5),

Name = c("Darshak","Roshan","Darshan","Jaivik","Sujal"),

Cpi = c(8.0,7.6,6.6,5.6,7.6)

)

cat("\nStudent Information\n\n")

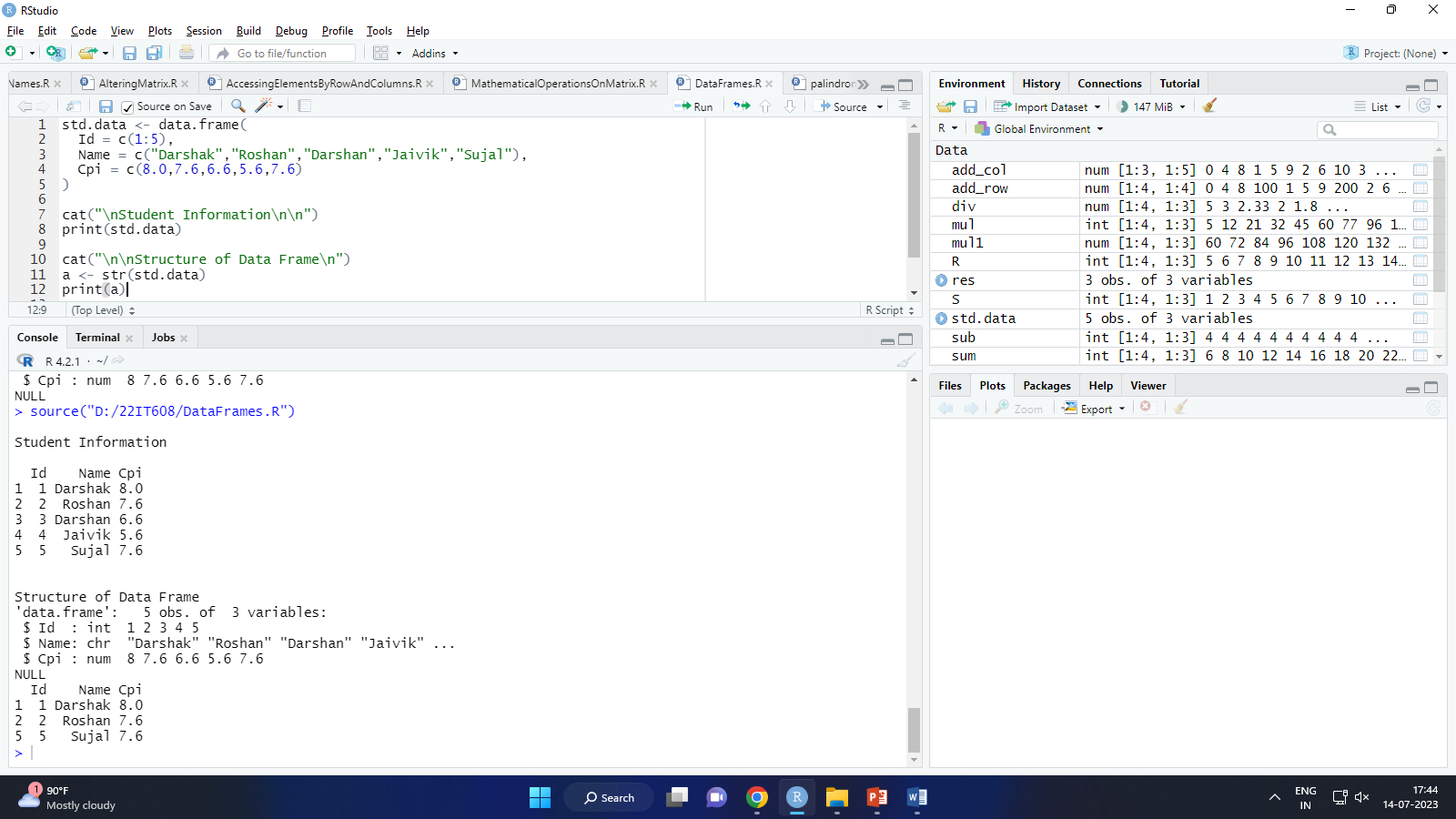
print(std.data)

cat("\n\nStructure of Data Frame\n")

a <- str(std.data)

print(a)

Output:



1. Write a program that feteches data of students who has more than 7.5 CPI.

Code:

std.data <- data.frame(

Id = c(1:5),

Name = c("Darshak","Roshan","Darshan","Jaivik","Sujal"),

Cpi = c(8.0,7.6,6.6,5.6,7.6)

)

cat("\nStudent Information\n\n")

print(std.data)

res <- std.data[std.data$Cpi > 7.5, ]

print(res)

Output:



1. Write a program that prints the summary of the data frame.

Code:

std.data <- data.frame(

Id = c(1:5),

Name = c("Darshak","Roshan","Darshan","Jaivik","Sujal"),

Cpi = c(8.0,7.6,6.6,5.6,7.6)

)

cat("\n\nSummary of the Data Frame\n\n")

print(summary(std.data))

Output:

