**Task -2 (10M)**

**Write a R script to illustrate the**

**a) Relational operators (5M)**

**b) Miscellaneous operators (5M)**

**on various datatypes.**

**Note: Inputs should be read from standard input and output should be sent to a separate file ‘my\_reslut.txt’.**

**CODE**

# Task -2

sink("my\_result.txt")

# relational operators

# for integer type

a<- as.integer(readline())

b<- as.integer(readline())

print(a>b)

print(a<b)

print(a<=b)

print(a>=b)

print(a==b)

print(a!=b)

#for numeric type

a<- as.numeric(readline())

b<- as.numeric(readline())

print(a>b)

print(a<b)

print(a<=b)

print(a>=b)

print(a==b)

print(a!=b)

#for character type

a<- as.character(readline())

b<- as.character(readline())

print(a>b)

print(a<b)

print(a<=b)

print(a>=b)

print(a==b)

print(a!=b)

#for logical type

a<- as.logical(readline())

b<- as.logical(readline())

print(a>b)

print(a<b)

print(a<=b)

print(a>=b)

print(a==b)

print(a!=b)

# for complex type

a<- as.complex(readline())

b<- as.complex(readline())

#print(a>b)

#print(a<b)

#print(a<=b)

#print(a>=b)

print(a==b)

print(a!=b)

#for raw type

a<- as.raw(readline())

b<- as.raw(readline())

print(a>b)

print(a<b)

print(a<=b)

print(a>=b)

print(a==b)

print(a!=b)

#for vectors

a<- scan()

b<- scan()

print(a>b)

print(a<b)

print(a==b)

print(a<=b)

print(a>=b)

print(a!=b)

# Miscellaneous operators

# " <- " operator

a<-16L

b<-1.5

c<-"Sagar"

d<-TRUE

e<-5-2i

f<-scan()

g<-charToRaw("String")

print(a)

print(b)

print(c)

print(d)

print(e)

print(f)

print(g)

# " -> " operator

12L->a

1.5->b

"Sagar"->c

TRUE->d

5-12i->e

scan()->f

charToRaw("String")->g

print(a)

print(b)

print(c)

print(d)

print(e)

print(f)

print(g)

# colon operator

a<- 1:10

print(i)

# %in% operator

x<- scan()

a<- 1

b<- 2

print(a %in% x)

print(b %in% x)

# %\*% operator

m<- matrix(scan(),nrow=2,ncol=2,byrow=TRUE)

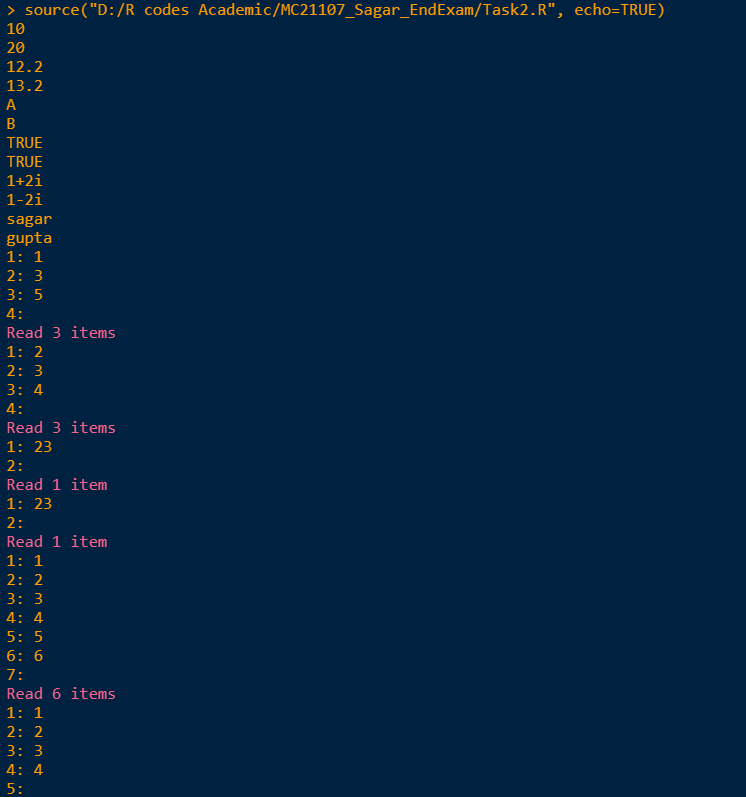
print(m)

n<- m%\*%t(m)

print(n)

sink()

**Standard Input**

****

**OUTPUT**

> # relational operators

> # for integer type

> a<- as.integer(readline())

> b<- as.integer(readline())

> print(a>b)

[1] FALSE

> print(a<b)

[1] TRUE

> print(a<=b)

[1] TRUE

> print(a>=b)

[1] FALSE

> print(a==b)

[1] FALSE

> print(a!=b)

[1] TRUE

> #for numeric type

> a<- as.numeric(readline())

> b<- as.numeric(readline())

> print(a>b)

[1] FALSE

> print(a<b)

[1] TRUE

> print(a<=b)

[1] TRUE

> print(a>=b)

[1] FALSE

> print(a==b)

[1] FALSE

> print(a!=b)

[1] TRUE

> #for character type

> a<- as.character(readline())

> b<- as.character(readline())

> print(a>b)

[1] FALSE

> print(a<b)

[1] TRUE

> print(a<=b)

[1] TRUE

> print(a>=b)

[1] FALSE

> print(a==b)

[1] FALSE

> print(a!=b)

[1] TRUE

> #for logical type

> a<- as.logical(readline())

> b<- as.logical(readline())

> print(a>b)

[1] FALSE

> print(a<b)

[1] FALSE

> print(a<=b)

[1] TRUE

> print(a>=b)

[1] TRUE

> print(a==b)

[1] TRUE

> print(a!=b)

[1] FALSE

> # for complex type

> a<- as.complex(readline())

> b<- as.complex(readline())

> #print(a>b)

> #print(a<b)

> #print(a<=b)

> #print(a>=b)

> print(a==b)

[1] FALSE

> print(a!=b)

[1] TRUE

> #for raw type

> a<- as.raw(readline())

> b<- as.raw(readline())

> print(a>b)

[1] FALSE

> print(a<b)

[1] FALSE

> print(a<=b)

[1] TRUE

> print(a>=b)

[1] TRUE

> print(a==b)

[1] TRUE

> print(a!=b)

[1] FALSE

> #for vectors

> a<- scan()

> b<- scan()

> print(a>b)

[1] FALSE FALSE TRUE

> print(a<b)

[1] TRUE FALSE FALSE

> print(a==b)

[1] FALSE TRUE FALSE

> print(a<=b)

[1] TRUE TRUE FALSE

> print(a>=b)

[1] FALSE TRUE TRUE

> print(a!=b)

[1] TRUE FALSE TRUE

> # Miscellaneous operators

>

> # " <- " operator

> a<-16L

> b<-1.5

> c<-"Sagar"

> d<-TRUE

> e<-5-2i

> f<-scan()

> g<-charToRaw("String")

> print(a)

[1] 16

> print(b)

[1] 1.5

> print(c)

[1] "Sagar"

> print(d)

[1] TRUE

> print(e)

[1] 5-2i

> print(f)

[1] 23

> print(g)

[1] 53 74 72 69 6e 67

> # " -> " operator

> 12L->a

> 1.5->b

> "Sagar"->c

> TRUE->d

> 5-12i->e

> scan()->f

> charToRaw("String")->g

> print(a)

[1] 12

> print(b)

[1] 1.5

> print(c)

[1] "Sagar"

> print(d)

[1] TRUE

> print(e)

[1] 5-12i

> print(f)

[1] 23

> print(g)

[1] 53 74 72 69 6e 67

> # colon operator

> a<- 1:10

> print(i)

[1] 8

> # %in% operator

> x<- scan()

> a<- 1

> b<- 2

> print(a %in% x)

[1] TRUE

> print(b %in% x)

[1] TRUE

> # %\*% operator

> m<- matrix(scan(),nrow=2,ncol=2,byrow=TRUE)

> print(m)

[,1] [,2]

[1,] 1 2

[2,] 3 4

> n<- m%\*%t(m)

> print(n)

[,1] [,2]

[1,] 5 11

[2,] 11 25

> sink()