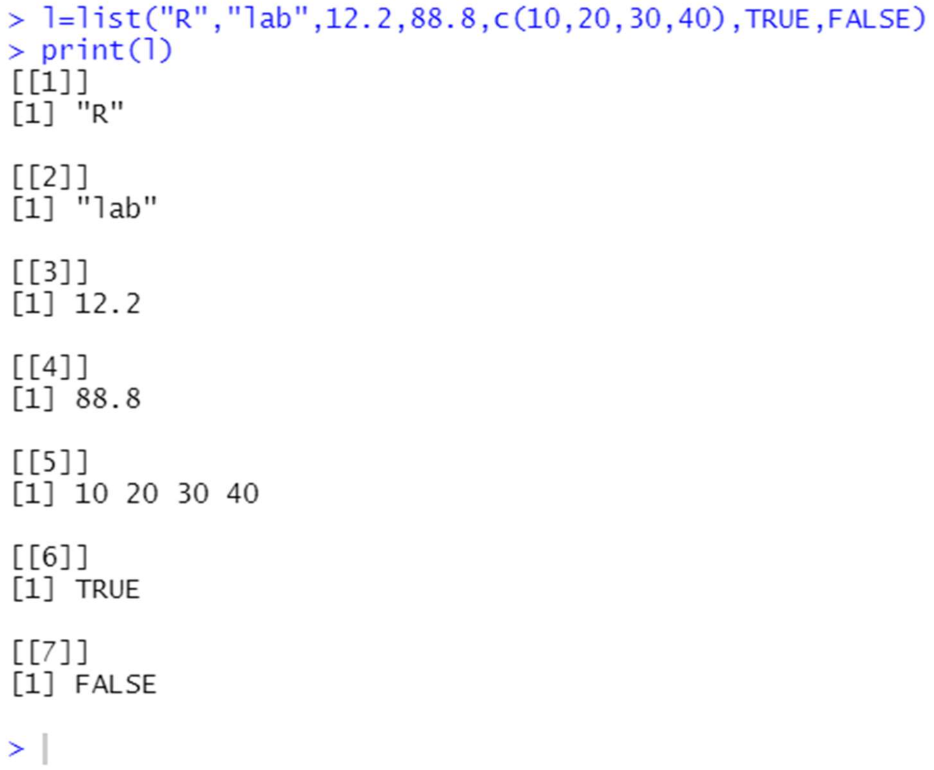
LISTS

Aim: Write an R program to sort a Vector in ascending and descending order.

Code:

```
l=list("R","lab",12.2,88.8,c(10,20,30,40),TRUE,FALSE)
print(l)
```

Output:



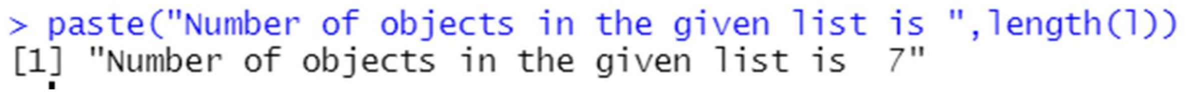
```
> l=list("R","lab",12.2,88.8,c(10,20,30,40),TRUE,FALSE)
> print(l)
[[1]]
[1] "R"

[[2]]
[1] "lab"

[[3]]
[1] 12.2

[[4]]
[1] 88.8

[[5]]
[1] 10 20 30 40

[[6]]
[1] TRUE

[[7]]
[1] FALSE

> |
```

Aim:-
Write an R program to count the number of objects in a list.

Code:

```
paste("Number of objects in the given list is ",length(l))
```

Output



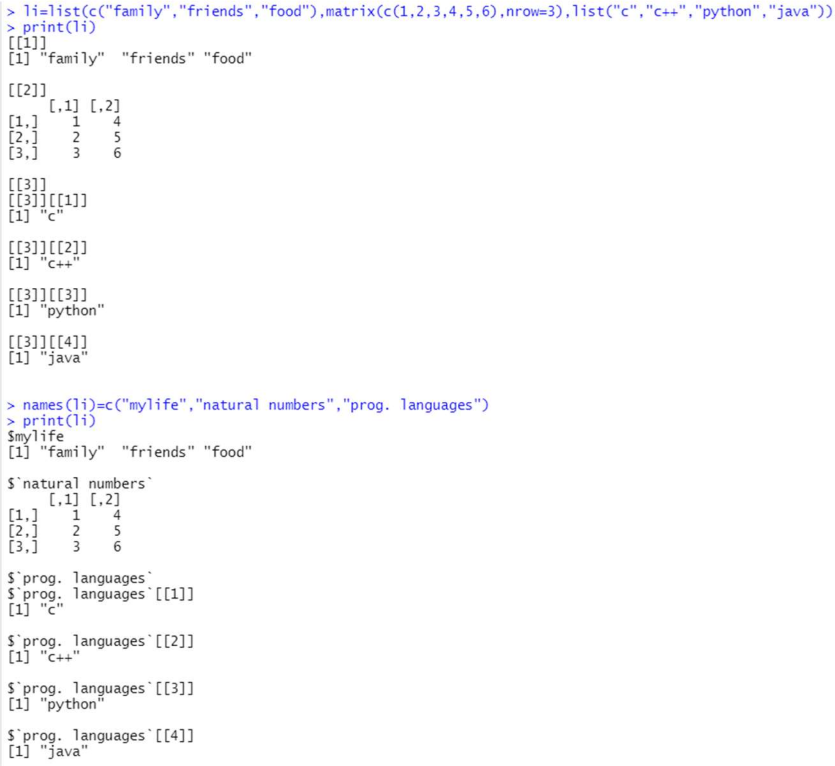
```
> paste("Number of objects in the given list is ",length(l))
[1] "Number of objects in the given list is  7"
```

Aim:- Write an R program to create a list containg a vector,a matrix and a list and give names to the elements in a list

Code:

```
li=list(c("family","friends","food"),matrix(c(1,2,3,4,5,6),nrow=3),list("c","c++","python","java"))
print(li)

names(li)=c("mylife","natural numbers","prog. languages")
print(li)
```

Output:



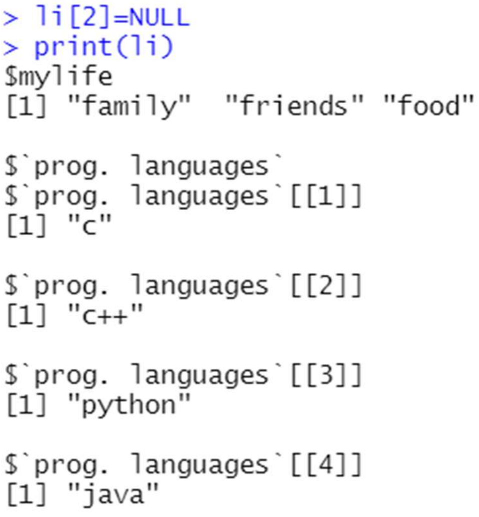
```
> li=list(c("family","friends","food"),matrix(c(1,2,3,4,5,6),nrow=3),list("c","c++","python","java"))
> print(li)
[[1]]
[1] "family"  "friends" "food"

[[2]]
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

[[3]]
[[3]][[1]]
[1] "c"

[[3]][[2]]
[1] "c++"

[[3]][[3]]
[1] "python"

[[3]][[4]]
[1] "java"


> names(li)=c("mylife","natural numbers","prog. languages")
> print(li)
$mylife
[1] "family"  "friends" "food"

$`natural numbers`
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

$`prog. languages`
$`prog. languages`[[1]]
[1] "c"

$`prog. languages`[[2]]
[1] "c++"

$`prog. languages`[[3]]
[1] "python"

$`prog. languages`[[4]]
[1] "java"
```

Aim:- Write an R program to create a list containg a vector,a matrix and a list and remove the 2nd element.

Code:

```
li[2]=NULL
print(li)
```

Output:



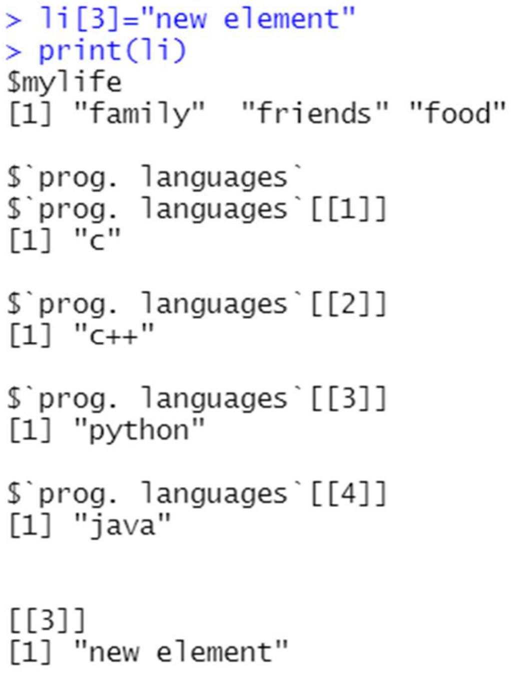
```
> li[2]=NULL
> print(li)
$mylife
[1] "family"  "friends" "food"

$`prog. languages`
$`prog. languages`[[1]]
[1] "c"

$`prog. languages`[[2]]
[1] "c++"

$`prog. languages`[[3]]
[1] "python"

$`prog. languages`[[4]]
[1] "java"
```

Aim:- Write an R program to create a list containg a vector,a matrix and list and element at the end of the list.

Code:

```
li[3]="new element"
print(li)
```

Output:



```
> li[3]="new element"
> print(li)
$mylife
[1] "family"  "friends" "food"

$`prog. languages`
$`prog. languages`[[1]]
[1] "c"

$`prog. languages`[[2]]
[1] "c++"

$`prog. languages`[[3]]
[1] "python"

$`prog. languages`[[4]]
[1] "java"


[[3]]
[1] "new element"
```

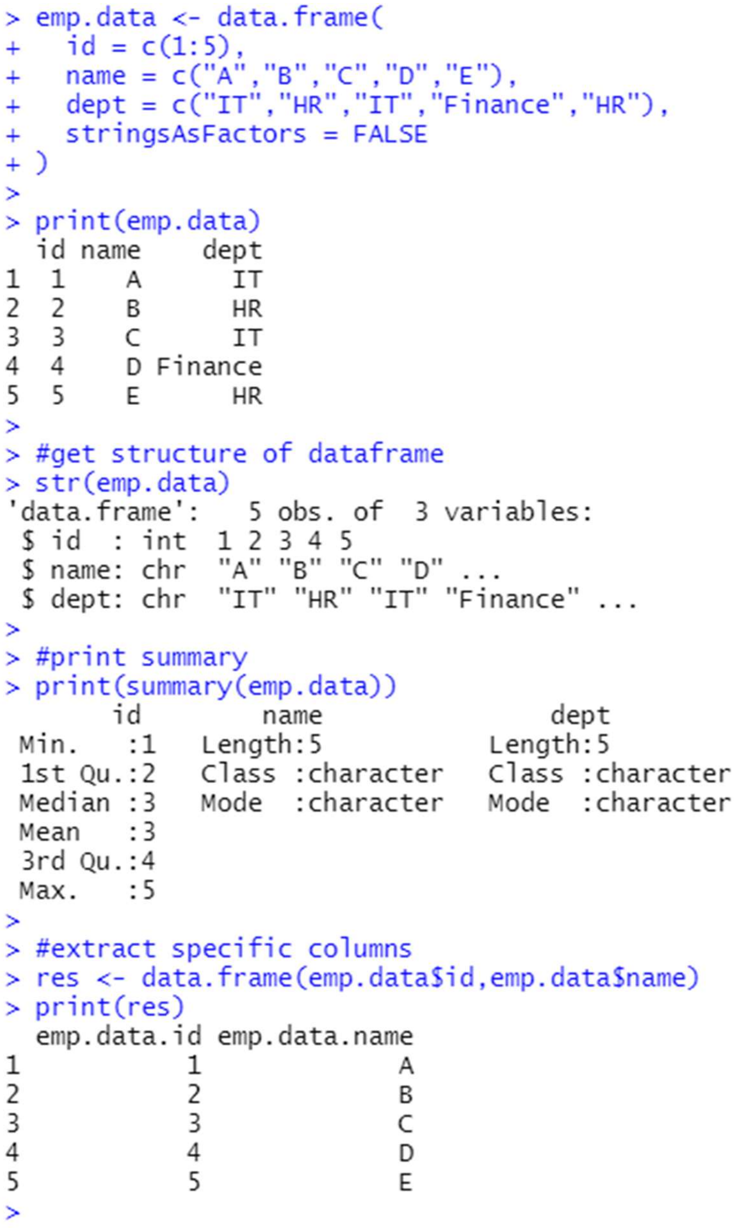
DATA FRAMES

Aim:- Write an R program to create a Data frame which contains details of 5 employees and display the data.

Code:

```
emp.data <- data.frame(
  id = c(1:5),
  name = c("A","B","C","D","E"),
  dept = c("IT","HR","IT","Finance","HR"),
  stringsAsFactors = FALSE
)

print(emp.data)

#get structure of dataframe
str(emp.data)

#print summary
print(summary(emp.data))

#extract specific columns
res <- data.frame(emp.data$id,emp.data$name)
print(res)
```

Output:



```
> emp.data <- data.frame(
+   id = c(1:5),
+   name = c("A","B","C","D","E"),
+   dept = c("IT","HR","IT","Finance","HR"),
+   stringsAsFactors = FALSE
+ )
>
> print(emp.data)
  id name    dept
1  1    A      IT
2  2    B      HR
3  3    C      IT
4  4    D Finance
5  5    E      HR
>
> #get structure of dataframe
> str(emp.data)
'data.frame':   5 obs. of  3 variables:
 $ id  : int  1 2 3 4 5
 $ name: chr  "A" "B" "C" "D" ...
 $ dept: chr  "IT" "HR" "IT" "Finance" ...
>
> #print summary
> print(summary(emp.data))
       id        name               dept
 Min.   :1   Length:5           Length:5
 1st Qu.:2   Class :character   Class :character
 Median :3   Mode  :character   Mode  :character
 Mean   :3
 3rd Qu.:4
 Max.   :5
>
> #extract specific columns
> res <- data.frame(emp.data$id,emp.data$name)
> print(res)
  emp.data.id emp.data.name
1           1             A
2           2             B
3           3             C
4           4             D
5           5             E
>
```

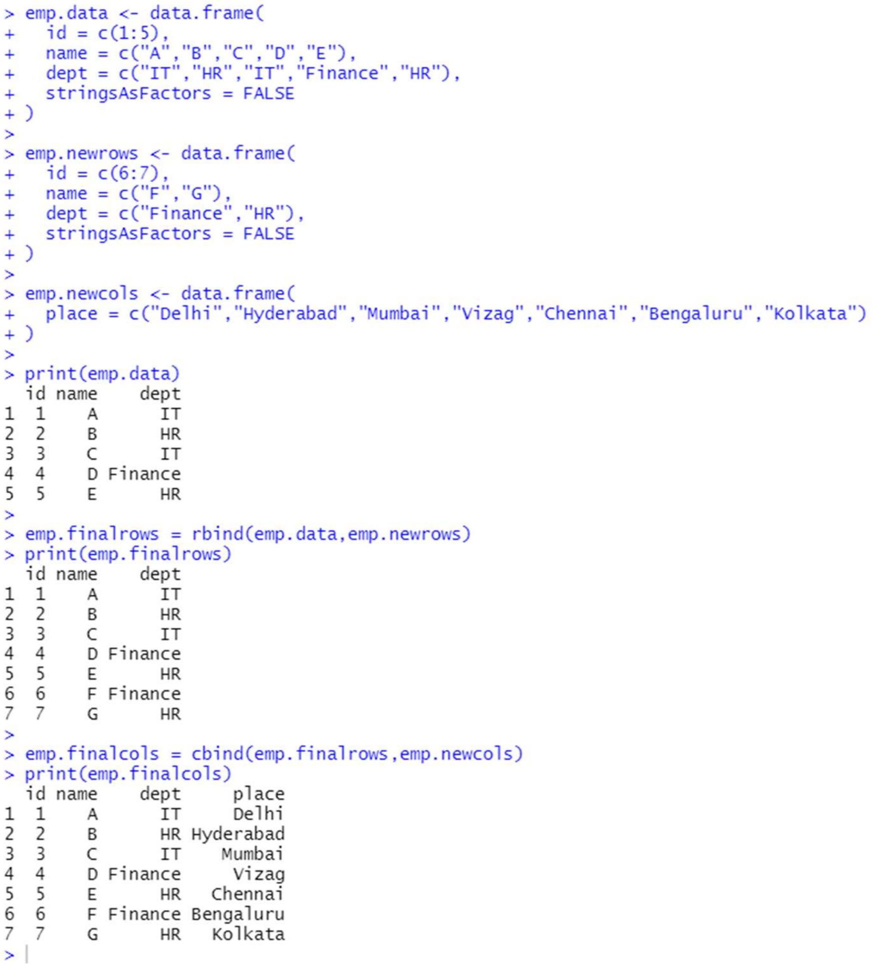
Aim:- Implement dataframes in R. Write a program to join columns and rows in a dataframe using cbind() and rbind() in R.

Code:

```
emp.data <- data.frame(
  id = c(1:5),
  name = c("A","B","C","D","E"),
  dept = c("IT","HR","IT","Finance","HR"),
  stringsAsFactors = FALSE
)

emp.newrows <- data.frame(
  id = c(6:7),
  name = c("F","G"),
  dept = c("Finance","HR"),
  stringsAsFactors = FALSE
)
```

```
emp.newcols <- data.frame(
  place = c("Delhi","Hyderabad","Mumbai","Vizag","Chennai","Bengaluru","Kolkata")
)

print(emp.data)

emp.finalrows = rbind(emp.data,emp.newrows)
print(emp.finalrows)

emp.finalcols = cbind(emp.finalrows,emp.newcols)
print(emp.finalcols)
```

Output:



```
> emp.data <- data.frame(
+   id = c(1:5),
+   name = c("A","B","C","D","E"),
+   dept = c("IT","HR","IT","Finance","HR"),
+   stringsAsFactors = FALSE
+ )
>
> emp.newrows <- data.frame(
+   id = c(6:7),
+   name = c("F","G"),
+   dept = c("Finance","HR"),
+   stringsAsFactors = FALSE
+ )
>
> emp.newcols <- data.frame(
+   place = c("Delhi","Hyderabad","Mumbai","Vizag","Chennai","Bengaluru","Kolkata")
+ )
>
> print(emp.data)
  id name    dept
1  1    A      IT
2  2    B      HR
3  3    C      IT
4  4    D Finance
5  5    E      HR
>
> emp.finalrows = rbind(emp.data,emp.newrows)
> print(emp.finalrows)
  id name    dept
1  1    A      IT
2  2    B      HR
3  3    C      IT
4  4    D Finance
5  5    E      HR
6  6    F Finance
7  7    G      HR
>
> emp.finalcols = cbind(emp.finalrows,emp.newcols)
> print(emp.finalcols)
  id name    dept     place
1  1    A      IT     Delhi
2  2    B      HR Hyderabad
3  3    C      IT    Mumbai
4  4    D Finance     Vizag
5  5    E      HR   Chennai
6  6    F Finance Bengaluru
7  7    G      HR   Kolkata
> |
```

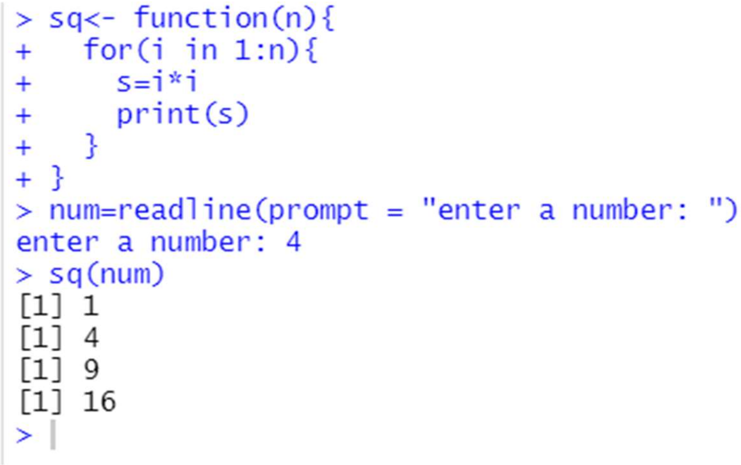
BASICS

Aim:Write an R program to create a function to print squares of a number in sequence.

Code:

```
sq<- function(n){
  for(i in 1:n){
    s=i*i
    print(s)
  }
}
num=readline(prompt = "enter a number: ")
sq(num)
```

Output:



```
> sq<- function(n){
+   for(i in 1:n){
+     s=i*i
+     print(s)
+   }
+ }
> num=readline(prompt = "enter a number: ")
enter a number: 4
> sq(num)
[1] 1
[1] 4
[1] 9
[1] 16
> 
```

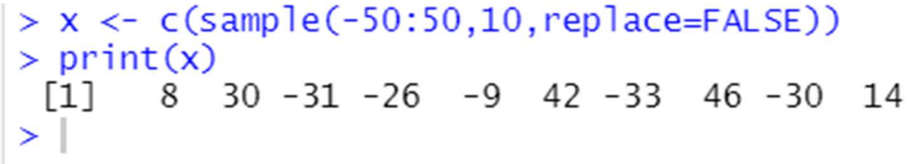
DATA TYPES

Aim:- Write an R program to create a vector which contains 10 random integer values between -50 and +50.

Code:

```
x <- c(sample(-50:50,10,replace=FALSE))
print(x)
```

Output:



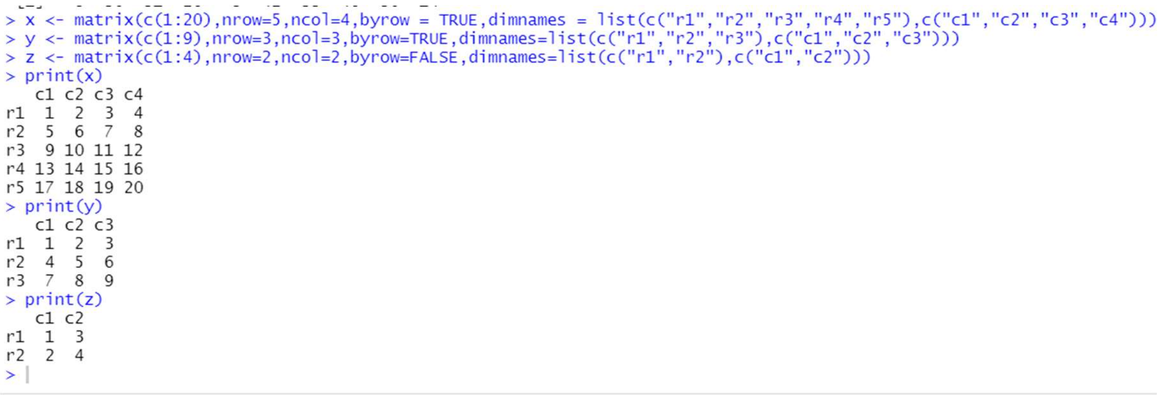
```
> x <- c(sample(-50:50,10,replace=FALSE))
> print(x)
 [1]   8  30 -31 -26  -9  42 -33  46 -30  14
> 
```

Aim:- Write an R program to create a 5 × 4 matrix, 3 × 3 matrix with labels and fill the matrix by rows and 2 × 2 matrix with labels and fill the matrix by columns.

Code:

```
x <- matrix(c(1:20),nrow=5,ncol=4,byrow = TRUE,dimnames =
list(c("r1","r2","r3","r4","r5"),c("c1","c2","c3","c4")))
y <- matrix(c(1:9),nrow=3,ncol=3,byrow=TRUE,dimnames=list(c("r1","r2","r3"),c("c1","c2","c3")))
z <- matrix(c(1:4),nrow=2,ncol=2,byrow=FALSE,dimnames=list(c("r1","r2"),c("c1","c2")))
print(x)
print(y)
print(z)
```

Output:



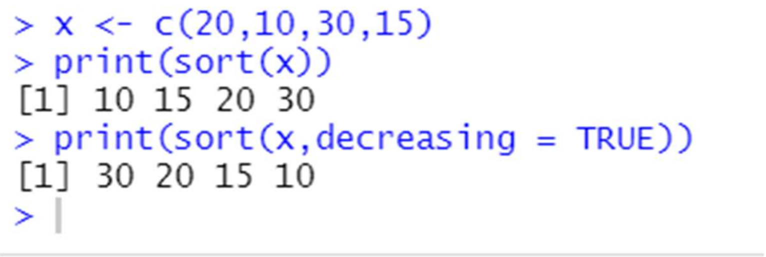
```
> x <- matrix(c(1:20),nrow=5,ncol=4,byrow = TRUE,dimnames = list(c("r1","r2","r3","r4","r5"),c("c1","c2","c3","c4")))
> y <- matrix(c(1:9),nrow=3,ncol=3,byrow=TRUE,dimnames=list(c("r1","r2","r3"),c("c1","c2","c3")))
> z <- matrix(c(1:4),nrow=2,ncol=2,byrow=FALSE,dimnames=list(c("r1","r2"),c("c1","c2")))
> print(x)
   c1 c2 c3 c4
r1  1  2  3  4
r2  5  6  7  8
r3  9 10 11 12
r4 13 14 15 16
r5 17 18 19 20
> print(y)
   c1 c2 c3
r1  1  2  3
r2  4  5  6
r3  7  8  9
> print(z)
   c1 c2
r1  1  3
r2  2  4
> |
```

Aim:- Write an R program to sort a Vector in ascending and descending order.

Code:

```
x <- c(20,10,30,15)
print(sort(x))
print(sort(x,decreasing = TRUE))
```

Output:



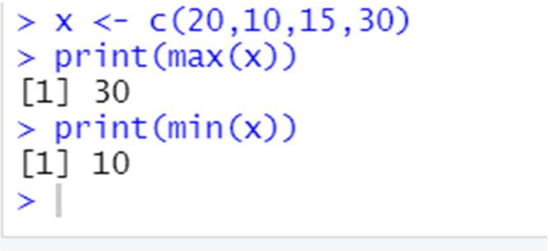
```
> x <- c(20,10,30,15)
> print(sort(x))
[1] 10 15 20 30
> print(sort(x,decreasing = TRUE))
[1] 30 20 15 10
> |
```

Aim:- Write an R program to find the maximum and the minimum value of a given vector.

Code:

```
x <- c(20,10,15,30)
print(max(x))
print(min(x))
```

Output:



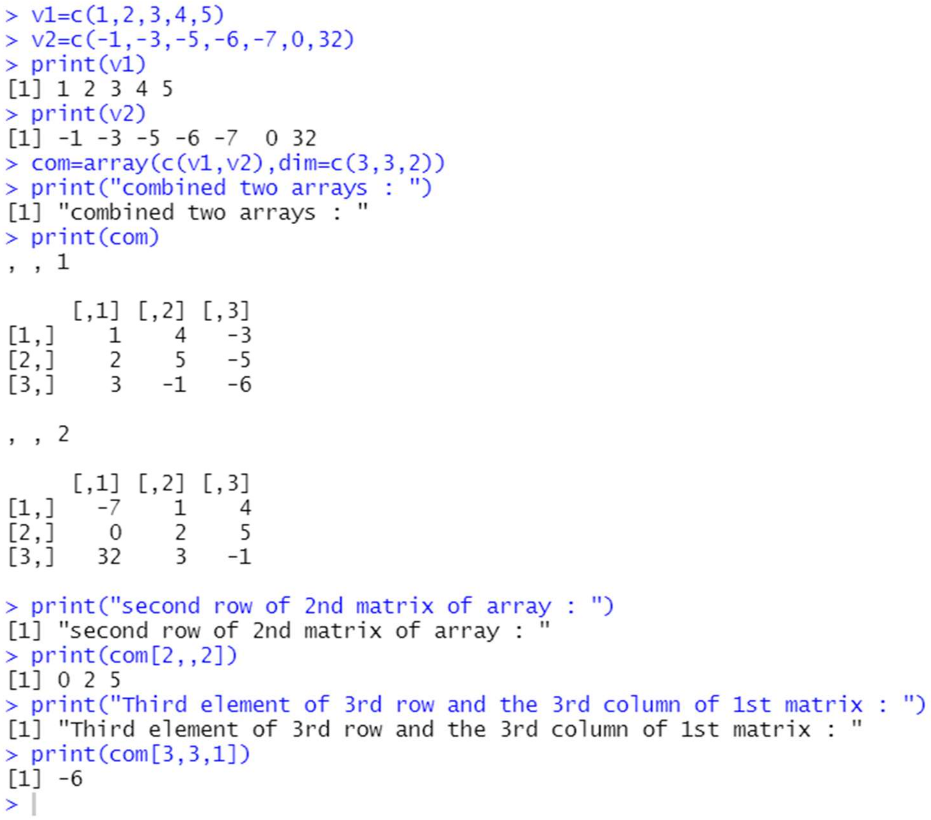
```
> x <- c(20,10,15,30)
> print(max(x))
[1] 30
> print(min(x))
[1] 10
> |
```

Aim:- Write an R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors. Print the second row of the second matrix of the array and the element in the 3rd row and 3rd column of the 1st matrix.

Code:

```
v1=c(1,2,3,4,5)
v2=c(-1,-3,-5,-6,-7,0,32)
print(v1)
print(v2)
com=array(c(v1,v2),dim=c(3,3,2))
print("combined two arrays : ")
print(com)
print("second row of 2nd matrix of array : ")
print(com[2,,2])
print("Third element of 3rd row and the 3rd column of 1st matrix : ")
print(com[3,3,1])
```

Output:



```
> v1=c(1,2,3,4,5)
> v2=c(-1,-3,-5,-6,-7,0,32)
> print(v1)
[1] 1 2 3 4 5
> print(v2)
[1] -1 -3 -5 -6 -7  0 32
> com=array(c(v1,v2),dim=c(3,3,2))
> print("combined two arrays : ")
[1] "combined two arrays : "
> print(com)
, , 1

     [,1] [,2] [,3]
[1,]    1    4   -3
[2,]    2    5   -5
[3,]    3   -1   -6

, , 2

     [,1] [,2] [,3]
[1,]   -7    1    4
[2,]    0    2    5
[3,]   32    3   -1

> print("second row of 2nd matrix of array : ")
[1] "second row of 2nd matrix of array : "
> print(com[2,,2])
[1] 0 2 5
> print("Third element of 3rd row and the 3rd column of 1st matrix : ")
[1] "Third element of 3rd row and the 3rd column of 1st matrix : "
> print(com[3,3,1])
[1] -6
> 
```

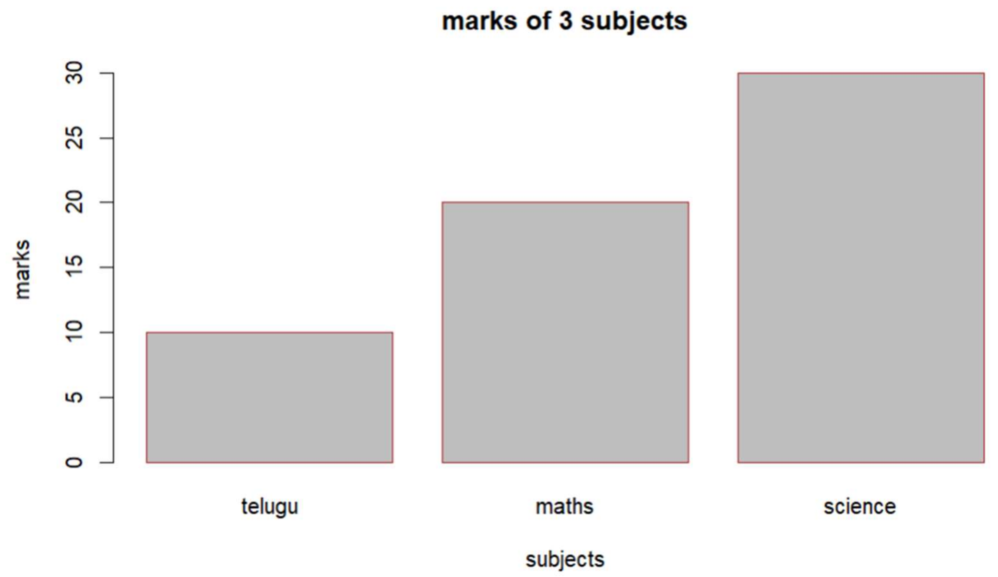
STATISTICS

Aim: Write an R program to create a simple bar plot of three subject’s marks, change the border color to brown and make inside bar lines as 90 degrees.

Code:

```
marks=c(10,20,30)
barplot(marks,main="marks of 3
subjects",xlab="subjects",ylab="marks",names.arg=c("telugu","maths","science"),angle=90,border
="brown",horiz="FALSE")
```

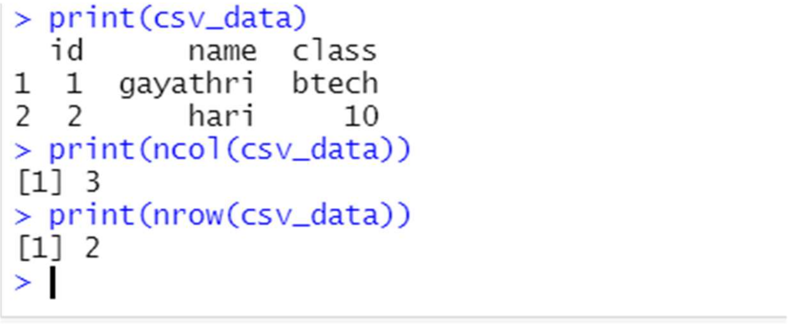
Output:



Aim: Write a program to read a csv file and analyze the data in the file in R.
Code:

```
setwd("G:/sem6/r_lab")

csv_data<-read.csv(file='sample.csv')
print(csv_data)
print(ncol(csv_data))
print(nrow(csv_data))
```

Output:

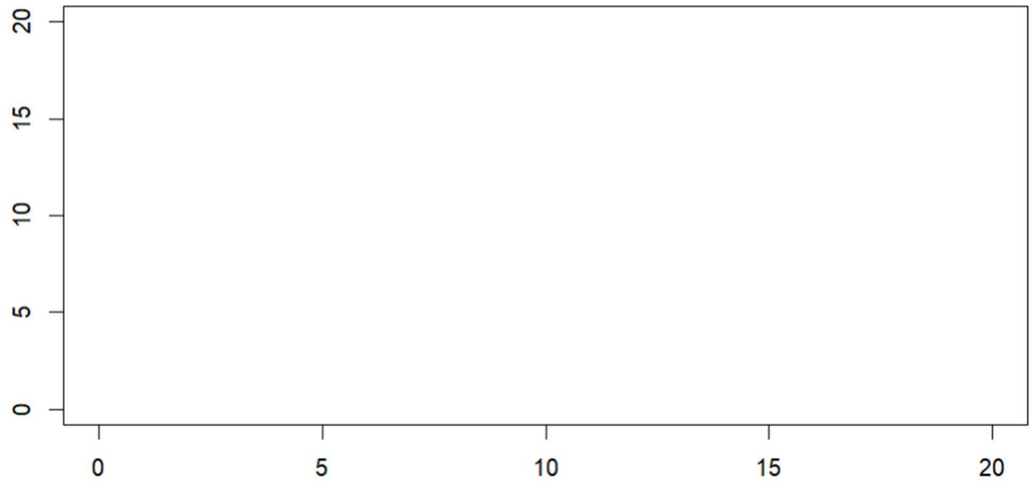


```
> print(csv_data)
  id      name  class
1  1  gayathri  btech
2  2      hari     10
> print(ncol(csv_data))
[1] 3
> print(nrow(csv_data))
[1] 2
> 
```

Aim: Write an R program to draw an empty pie chart and empty plots specify the axes limits of the graph
Code:

```
plot.new()
plot(1,type="n",xlab="",ylab="",xlim=c(0,20),ylim=c(0,20))
```

Output:

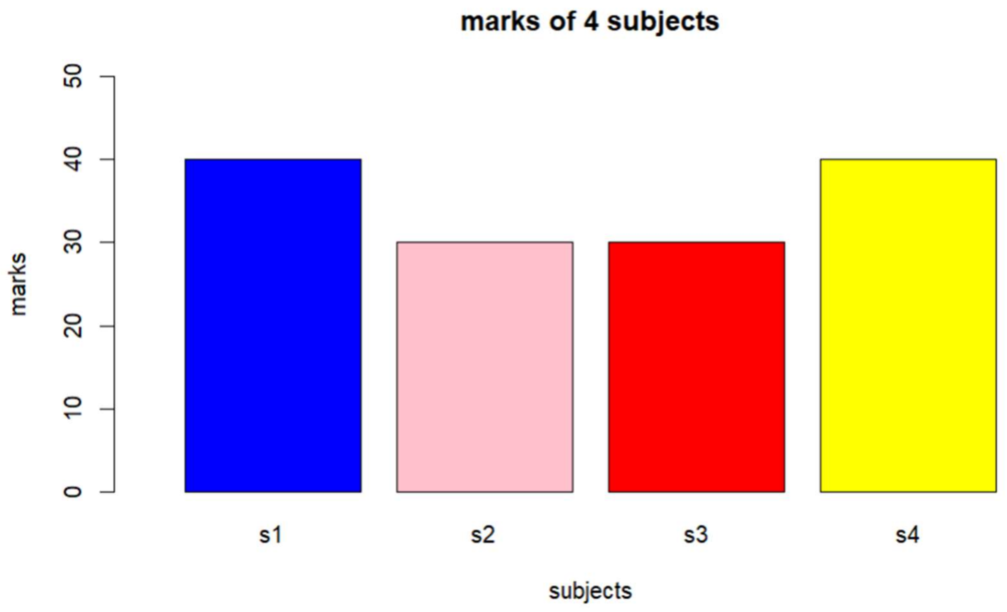


Aim: Write an R program to create a simple bar plot of four subject’s registered, assign the colors to each bar and assign the limit to x-axis as c(0,5) and y-axis as c(0,50).

Code:

```
marks<-c(40,30,30,40)
clr<-c("blue","pink","red","yellow")
barplot(marks,main="marks of 4 subjects",xlab="subjects",ylab="marks",xlim=c(0,5),ylim =
c(0,50),col = clr,names.arg=c("s1","s2","s3","s4"))
```

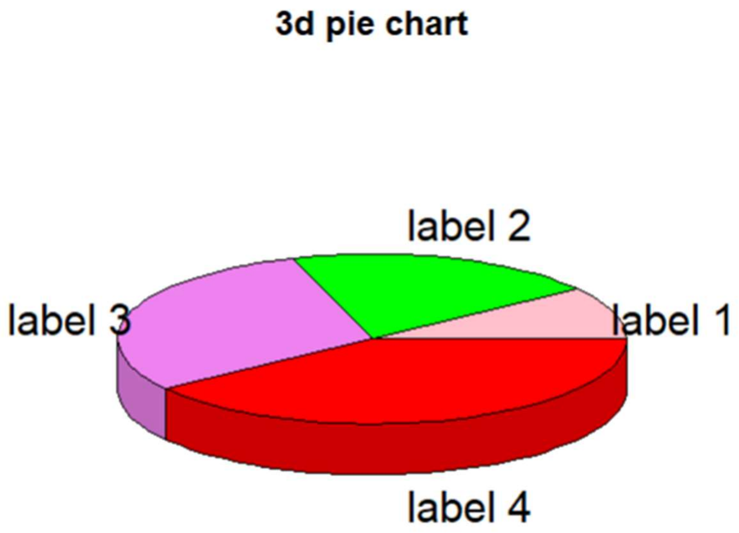
Output:



Aim: Write an R program to create a simple 3D pie chart, assign color and labels to each part.
Code:

```
library(plotrix)
x<-c(10,20,30,40)
pie3D(x,main="3d pie chart",labels=c("label 1","label 2","label 3","label
4"),col=c("pink","green","violet","red"))
```

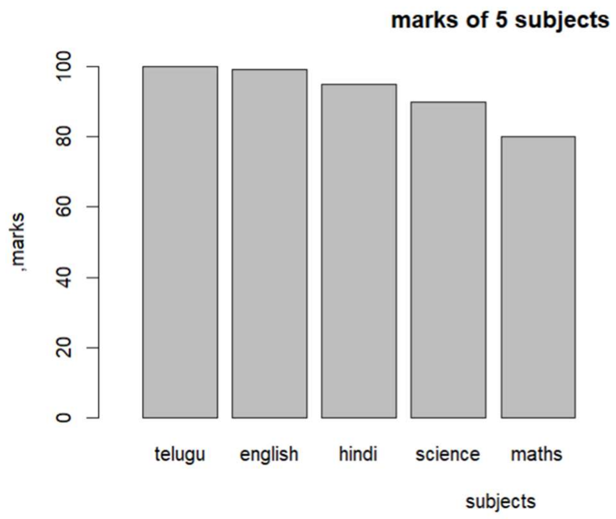
Output:



Aim: Write an R program to create a simple bar plot of five subject's marks.
Code:

```
marks<-c(100,99,95,90,80)
subjects<-c("telugu","english","hindi","science","maths")
barplot(marks,names.arg = subjects,xlab="subjects",ylab=",marks",main="marks of 5
subjects",xlim=c(0,10),ylim=c(0,100))
```

Output:

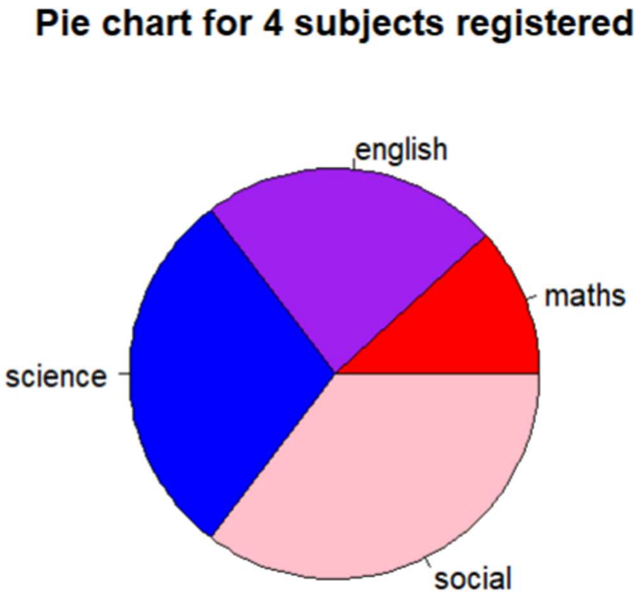


Aim: Write an R program to create a simple pie chart of four subjects registered, assign the colors to each block and display in anti-clockwise direction.

Code:

```
subjects<-c(20,40,50,60)
colors<-c("red","purple","blue","pink")
pie(subjects,main="Pie chart for 4 subjects registered",labels =
c("maths","english","science","social"),col = colors,clockwise = FALSE)
```

Output:

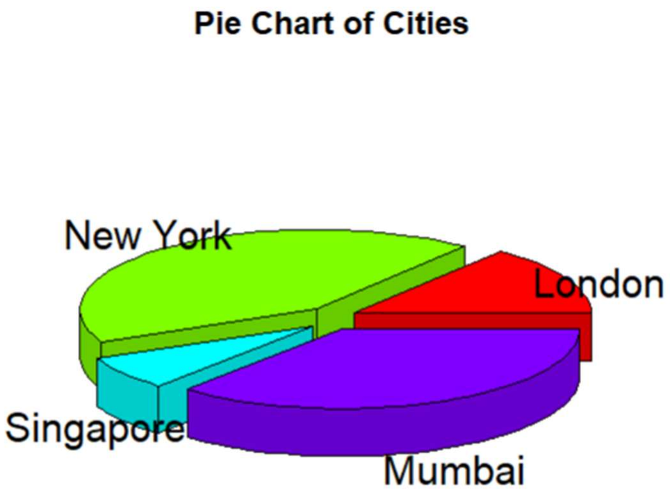


Aim: Write an R program to create a simple 3D pie chart, assign title to the chart and also split each part.

Code:

```
library(plotrix)
parts<-c(40,30,30,40)
label<-c("India","US","Newyork","London")
pie3D(parts,main="pie chart representing talents in each country",labels =
label,col=c("yellow","pink","purple","red"),explode=0.1)
```

Output:

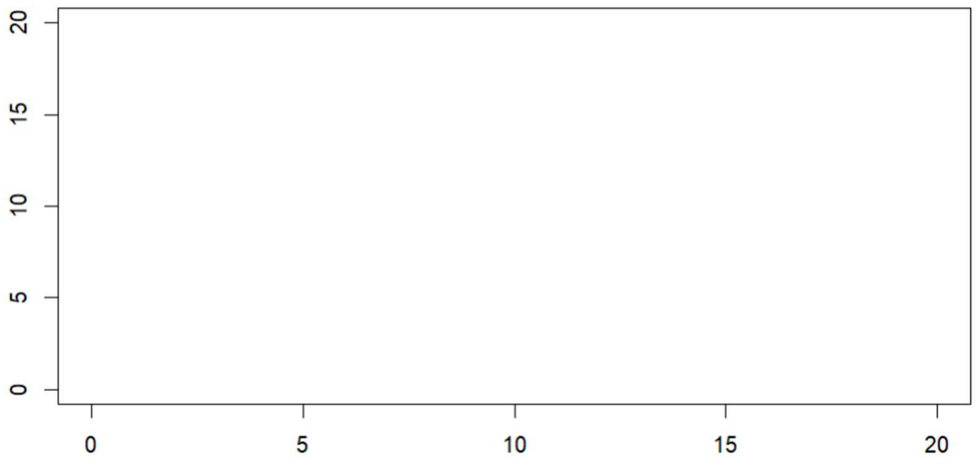


Aim: Write an R program to draw an empty bar plot and empty plots specify the axes limits of the graph.

Code:

```
plot.new()
plot(1,type="n",xlab="",ylab="",xlim=c(0,20),ylim=c(0,20))
```

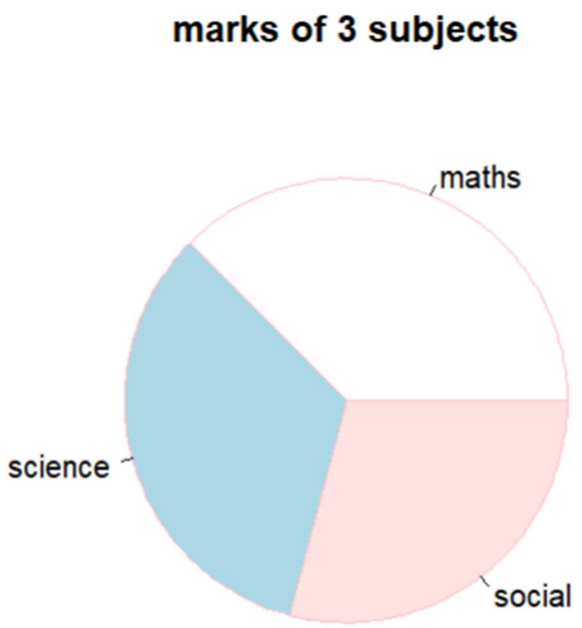
Output:



Aim: Write an R program to create a simple pie chart of three subject’s marks, change the border color to pink and make inside bar lines as 60 degrees.
Code:

```
marks<-c(90,80,70)
pie(marks,labels = c("maths","science","social"),main="marks of 3 subjects",angle = 60,border =
"pink")
```

Output:


marks of 3 subjects
maths
science
social