

1. Printing hello world
myString <- "Hello World"
myString

2. To get today's date and current time

```
cat (date())
```

3. To get the version of R on which you are working
print(R.version.string)

4. To calculate area of a circle, square and triangle

```
r <- 5.9  
area = 3.14*r*r  
cat("The area of the circle is ",area)  
a <- 5  
area_of_square = a^2  
cat ("Area of square", area_of_square)  
b = 4  
area_of_rect = 0.8* (a*b)  
cat ("Area of rect", area_of_rect)
```

5. To find out whether a number is Positive, Zero or Negative

```
a = -22  
if(a>=0){ print (positive)} else {print ("Negative")}
```

6. To convert kilometers to miles

```
a = 8  
miles = a/1.609  
cat ("The miles are", miles)
```

7. To convert Celsius to Fahrenheit

```
c = 5.23  
fer = {(c/5)*9}+32  
cat("The fahrenheit are",fer)
```

8. To find out whether a number is Positive, Zero or Negative. Ask user to give the input number.

```
num <- as.integer(readline(prompt = "Enter number:"))  
if(num>=0){ if (num==0) {print ("Zero")} else{print ("Positive")}} else {print ("Negative")}
```

9. To find whether a string is palindrome. Ask user to give the input a string.

```
str = readline (prompt = "Enter string:")  
splits <- strsplit(str, " ")[[1]]
```

```
str_rev <- rev(splits)
pal <- paste(str_rev, collapse = " ")
if(str == pal){ cat("palindrome string") } else{cat("Not a palindrome string")}
```

10.To check if a year is leap year

```
a=2000
if ((a %% 400) == 0)
{ print ("Leap Year")
  } else if ((a %% 100) == 0 )
  {print ("Not a Leap Year")
   } else if ((a %% 4) == 0 )
  {print ("LEAP YEAR")
   } else {print("Not a Leap Year")}
```

11.to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.

```
cat ("Sequence of numbers from 20 to 50:",seq (20, 50) , "\n")
cat ("Mean of numbers from 20 to 60:", mean(seq (20, 60)), "\n")
cat ("sum of numbers from 51 to 91:" ,sum (seq (51, 91)), "\n")
```

12.to get the first 10 Fibonacci numbers

```
num = 10
n1 <- 0; n2 -> 1
cnt = 2
if(num == 1) {
  cat ("Fibonacci sequence:\n")
  print (n1)
} else{
  cat ("Fibonacci sequence:\n")
  print (n1)
  print (n2)
  while(cnt < num) {
    nf = n1 + n2
    print(nf)
    n1=n2
    n2 = nf
    cnt = cnt + 1
  }
}
```

13.to extract first 10 english letter in lower case and last 10 letters in upper case and extract letters between 22nd to 24th letters in upper case

```
cat ("First 10 letters in lower case:",head(letters,10),"\n")
cat ("Last 10 letters in upper case:",tail (LETTERS, 10),"\n")
cat ("Letters between 22nd to 24th letters in upper case:",head(LETTERS[22:24]),"\n")
```

14.to get the unique elements of a given string and unique numbers of

vector.

```
nums = c( 1, 2, 3, 4, 5, 3, 2, 1)
print(unique(nums))
```

15.to create three vectors a,b,c with 3 integers. Combine the three vectors to become a 3×3 matrix where each column represents a vector. Print the content of the matrix.

```
a = c(1:3)
b =c(4,7,6)
c =c(7:(8+1))
r = matrix(c(a , b, c), nrow = 3, ncol = 3)
print(r)
```

16.to create a matrix taking a given vector of numbers as input. Display the matrix.

```
print ("GDP of Countries: \n")
print (matrix(c(25.2,20.5,19.6,22.2,25.8,24.9), nrow = 2,byrow = TRUE,
              dimnames = list(c("India", "U. S. "), c("2015", "2016", "2017")) ))
```

17.to access the element at 3rd column and 2nd row, only the 3rd row and only the 4th column of a given matrix.

```
r = matrix(c(c(1:3) , c(4:6), c(7:9), c(10:12)), nrow = 3, ncol = 4)
print(r)
cat ("\nThe element at 3rd column and 2nd row",r[2,3])
cat ("\nThe 3rd row",r[3,])
cat ("\nThe 4th column",r[,4])
```

18.to create two 2x3 matrix and add, subtract, multiply and divide the matrixes.

```
m1 = matrix(c(7:12), nrow = 2, ncol = 3)
m2 = matrix(c(1:6), nrow = 2, ncol = 3)
cat("\nMatrix1:")
print (m1)
cat ("\nMatrix2:")
print (m2)
m3 = m1 + m2
cat ("\nMatrix Addition:")
print (m3)
m3 = m1 - m2
cat ("\nMatrix Subtraction:")
print (m3)
m3 = m1 * m2
cat("\nMatrix Multipliction: ")
```

```
print (m3)
m3 = m1 / m2
cat("\nMatrix divition: ")
print (m3)
```

19.to extract the submatrix whose rows have column value > 7 from a given matrix.

```
m1 = matrix(c(1:9), nrow = 3, ncol = 3, byrow = TRUE)
cat ("The original Matrix: \n")
print (m1)
submatrix = m1 [m1 [, 3] > 7]
cat ("The submatrix: \n")
print (submatrix)
```

20.to find row and column index of maximum and minimum value in a given matrix.

```
m1= matrix(c(c(7:12), c(1:6)), nrow = 4, ncol = 3, byrow = TRUE)
print(m1)
min_i=which(m1 == min(m1), arr.ind = TRUE)
max_i=which(m1 == max(m1), arr.ind = TRUE)
cat ("The min value of matrix at index:\n")
print(min_i)
cat ("The max value of matrix at index:\n")
print(max_i)
```

21.to concatenate two given matrices of same column but different rows.

```
m1= matrix (c(1:8), nrow = 4, ncol = 2, byrow = TRUE)
m2= matrix (c(9:14), nrow = 3, ncol = 2, byrow = TRUE)
m3=rbind (m1, m2)
cat ("Concatenated Matrix: \n")
print(m3)
```

22.to find Sum, Mean and Product of a Vector, ignore element like NA or NaN.

```
v1 <- c(1,2,6,78,NaN,NA, 42,4, NA, 5 , NaN)
print (v1)
cat ("The sum of vector elements:",sum(v1, na.rm = TRUE))
cat ("The mean of vector elements:", mean (v1, na.rm = TRUE))
cat ("The product of vector elements:",prod(v1, na.rm= TRUE))
```

23.to sort a Vector in ascending and descending order.

```
v1 <- c(-1,2.5,6.47, -78, NaN, NA, 42, 4.22, 5)
print (v1)
cat ("\nThe vector in ascending order: \n")
print (sort(v1))
cat ("\nThe vector in descending order: \n")
print(sort (v1,decreasing = TRUE))
```