

```

# 1. Create a vector using : operator
# a. Sequence from -5 to 5. Write the R code and its output.
num <- -5:5
num

# Output
#[1] -5 -4 -3 -2 -1  0  1  2  3  4  5

# Describe its output:
# It created a section of vector that starts from -5 upto 5

#b. x <- 1:7. What will be the value of x?

x <- 1:7
x

# Output
#[1] 1 2 3 4 5 6 7
# The value of x is a sequence of integers from 1 to 7.

#2.* Create a vector using seq() function
# a. seq(1, 3, by=0.2) # specify step size

seq(1, 3, by =0.2)

# Output
# [1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0

#Describe the output
#The sequence starts at 1 and it adds by 0.2 until it reach 3

# 3. A factory has a census of its workers. There are 50 workers in total.

workers_age <- c(34, 28, 22, 36, 27, 18, 52, 39, 42, 29, 35, 31, 27,
                22, 37, 34, 19, 20, 57, 49, 50, 37, 46, 25, 17, 37, 43, 53, 41, 51, 35,
                24,33, 41, 53, 40, 18, 44, 38, 41, 48, 27, 39, 19, 30, 61, 54, 58, 26,
                18)

# a. Access 3rd element, what is the value?
workers_age[3]
# The value would be 22
#b. Access 2nd and 4th element, what are the values?
workers_age[c(2, 4)]

# The value would be 28 for 2nd element and 36 for 4th element

# c. Access all but the 1st element is not included. Write the R code and its output.

workers_age[2:50]

# Output
# [1] 28 22 36 27 18 52 39 42 29 35 31 27 22 37 34 19 20 57 49 50 37
# [22] 46 25 17 37 43 53 41 51 35 24 33 41 53 40 18 44 38 41 48 27 39
# [43] 19 30 61 54 58 26 18

# 4. *Create a vector x <- c("first"=3, "second"=0, "third"=9). Then named the vector.

x <- c("first"=3, "second"=0, "third"=9)

x[c("first", "third")]

# a. Print the results. Then access x[c("first", "third")].

# Output:

```

```

# first third
#   3       9

# Describe the output:
# It created a named vector where columns are first and third, and rows is 3 and 9

# b. Write the code and its output.

x

# Output
# first second  third
#     3       0      9

# 5. Create a sequence x from -3:2.
x <- -3:2

# a. Modify 2nd element and change it to 0;
x[2] <- 0
x

# Describe the output:
# It replaces the index 2 element which is -2 and replace it to 0

# b. Write the code and its output.
# Output
# [1] -3  0 -1  0  1  2

# 6. *The following data shows the diesel fuel purchased by Mr. Cruz.

# a. Create a data frame for month, price per liter (php) and purchase-quantity (liter).
# Write the codes.

month <- c("Jan", "Feb", "March", "Apr", "May", "June")
price_per_liter <- c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00)
purchase_quantity <- c(25, 30, 40, 50, 10, 45)
fuel_data <- data.frame(month, price_per_liter, purchase_quantity)

# b. What is the average fuel expenditure of Mr. Cruz from Jan to June? Note: Use
# weighted.mean(liter, purchase)

average_price <- weighted.mean(fuel_data$price_per_liter, fuel_data$purchase_quantity)

average_price

# Output [1] 59.2625

# 7. R has actually lots of built-in datasets. For example, the rivers data "gives the lengths
# (in miles) of 141 "major" rivers in North America, as compiled by the US Geological Survey".

# a. Type "rivers" in your R console. Create a vector data with 7
# elements, containing the number of elements (length) in rivers,
# their sum mean, median, variance, standard deviation, minimum and maximum

data <- c(length(rivers), sum(rivers), mean(rivers), median(rivers), var(rivers),
          sd(rivers), min(rivers), max(rivers))

data

# b. What are the results?
# The results are length is 141.0000 Sum is 83357.0000 mean is 591.1844

```

```
# Median is 425.0000 var is 243908.4086 sd is 493.8708 min is 135.0000 and max is 3710.0000
```

```
# c. Write the code and its outputs.
```

```
data <- c(length(rivers), sum(rivers), mean(rivers), median(rivers), var(rivers),
          sd(rivers), min(rivers), max(rivers))
```

```
data
```

```
# 8. The table below gives the 25 most powerful celebrities and their annual pay as ranked
# by the editions of Forbes magazine and as listed on the Forbes.com website.
```

```
# a. Create vectors according to the above table. Write the codes.
```

```
power_ranking <- 1:25
```

```
celebrity_name <- c(
  "Tom Cruise", "Rolling Stones", "Oprah Winfrey", "U2", "Tiger Woods",
  "Steven Spielberg", "Howard Stern", "50 Cent", "Cast of the Sopranos",
  "Dan Brown", "Bruce Springsteen", "Donald Trump", "Muhammad Ali",
  "Paul McCartney", "George Lucas", "Elton John", "David Letterman",
  "Phil Mickelson", "J.K. Rowling", "Bradd Pitt", "Peter Jackson",
  "Dr. Phil McGraw", "Jay Leno", "Celine Dion", "Kobe Bryant"
)
```

```
pay <- c(
  67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55,
  40, 233, 34, 40, 47, 75, 25, 39, 45, 32, 40, 31
)
```

```
# b. Modify the power ranking and pay of J.K. Rowling. Change power ranking to 15 and
# pay to 90. Write the codes and its output.
```

```
power_ranking[19] <- 15
```

```
pay[19] <- 90
```

```
celebrity_data <- data.frame(
  Ranking = power_ranking,
  Name = celebrity_name,
  Pay = pay
)
```

```
print(celebrity_data)
```

```
# C Interpret the data
```

```
# The data shows a list of ranking, names and their pays.
```

```
# It also shows that there are total of 25 people and the Highest pay is Steven SPEILBERG
```

```
# It means that although Speilberg was the highest pay, it doesn't guaranteed him to be the top 1
ranking
```