1. The table below shows the data about shoe size and height. Create a data frame.

```
shoe_size <- c(6.5, 9.0, 8.5, 8.5, 10.5, 7.0, 9.5, 9.0, 13.0, 7.5, 10.5, 8.5, 12.0, 10.5, 13.0, 11.5, 8
height <-  c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 70.0, 71.0, 72.0, 64.0, 74.5, 67.0, 71.0, 71.0, 77.0, 
gender <- c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F", "M", "F", "M", "M", "M", "M", "F", "F", "

frame1 <- data.frame(Shoesize = shoe_size, Height = height, Gender = gender)
frame1
```

```
##    Shoesize Height Gender
## 1       6.5   66.0      F
## 2       9.0   68.0      F
## 3       8.5   64.5      F
## 4       8.5   65.0      F
## 5      10.5   70.0      M
## 6       7.0   64.0      F
## 7       9.5   70.0      F
## 8       9.0   71.0      F
## 9      13.0   72.0      M
## 10      7.5   64.0      F
## 11     10.5   74.5      M
## 12      8.5   67.0      F
## 13     12.0   71.0      M
## 14     10.5   71.0      M
## 15     13.0   77.0      M
## 16     11.5   72.0      M
## 17      8.5   59.0      F
## 18      5.0   62.0      F
## 19     10.0   72.0      M
## 20      6.5   66.0      F
## 21      7.5   64.0      F
## 22      8.5   67.0      M
## 23     10.5   73.0      M
## 24      8.5   69.0      F
## 25     10.5   72.0      M
## 26     11.0   70.0      M
## 27      9.0   69.0      M
## 28     13.0   70.0      M
```

b. Create a subset by males and females with their corresponding shoe size and height. What its result? Show the R scripts.

```
female <- subset(frame1, Gender == "F", select = c(Shoesize, Height))
female
```

```
##    Shoesize Height
## 1       6.5   66.0
## 2       9.0   68.0
## 3       8.5   64.5
## 4       8.5   65.0
## 6       7.0   64.0
## 7       9.5   70.0
## 8       9.0   71.0
## 10      7.5   64.0
## 12      8.5   67.0
## 17      8.5   59.0
```

```
## 18      5.0   62.0
## 20      6.5   66.0
## 21      7.5   64.0
## 24      8.5   69.0
```

```
male <- subset(frame1, Gender == "M", select = c(Shoesize, Height))
male
```

```
##    Shoesize Height
## 5      10.5   70.0
## 9      13.0   72.0
## 11     10.5   74.5
## 13     12.0   71.0
## 14     10.5   71.0
## 15     13.0   77.0
## 16     11.5   72.0
## 19     10.0   72.0
## 22      8.5   67.0
## 23     10.5   73.0
## 25     10.5   72.0
## 26     11.0   70.0
## 27      9.0   69.0
## 28     13.0   70.0
```

c. Find the mean of shoe size and height of the respondents. Write the R scripts and its result.

```
meanshoe <- mean(frame1$Shoesize)
meanshoe
```

```
## [1] 9.410714
```

```
meanheight <- mean(frame1$Height)
meanheight
```

```
## [1] 68.57143
```

d. Is there a relationship between shoe size and height? Why?

```
relation <- cor(frame1$Shoesize, frame1$Height)
relation
```

```
## [1] 0.7766089
```

2. Construct character vector months to a factor with factor() and assign the result to factor_months_vector. Print out factor_months_vector and assert that R prints out the factor levels below the actual values.

```
months <- c("March", "April", "January", "November", "January", "September", "October", "September", "N

fmonths <- factor(months)
fmonths
```

```
##  [1] March     April     January   November  January   September October
##  [8] September November  August    January   November  November  February
## [15] May       August    July      December  August    August    September
## [22] November  February  April
## 11 Levels: April August December February January July March May ... September
```

3. Then check the summary() of the months_vector and factor_months_vector. Interpret the results of both vectors. Are they both equally useful in this case?

```r
smonths <- summary(months)
sfmonths <- summary(fmonths)

sfmonths
```

```
##     April    August  December  February   January      July     March       May
##         2         4         1         2         3         1         1         1
##  November   October September
##         5         1         3
```

```r
smonths
```

```
##    Length     Class      Mode
##        24 character character
```

4. Create a vector and factor for the table below.

```r
factor_data <- c(rep("East", 1), rep("West", 4), rep("North", 3))
new_order_data <- factor(factor_data, levels = c("East", "West", "North"))

new_order_data
```

```
## [1] East  West  West  West  West  North North North
## Levels: East West North
```

5. Enter the data below in Excel with file name = import_march.csv

a. Import the excel file into the Environment Pane using read.table() function. Write the code.
b. View the dataset. Write the R scripts and its result.

```r
data <- read.table("import_march.csv", header = TRUE, sep = ",")
data
```

```
##   Students Strategy.1 Strategy.2 Strategy.3
## 1     Male          8         10          8
## 2                   4          8          6
## 3                   0          6          4
## 4   Female         14          4         15
## 5                  10          2         12
## 6                   6          0          9
```

6. Full Search

a. Create an R Program that allows the User to randomly select numbers from 1 to 50. Then display the chosen number. If the number is beyond the range of the selected choice, it will have to display a string "The number selected is beyond the range of 1 to 50". If number 20 is inputted by the User, it will have to display "TRUE", otherwise display the input number.

```r
exhaustive_search <- function() {
  chosen_number <- as.integer(readline(prompt = "Select a number between 1 and 50: "))

  if (is.na(chosen_number)) {
    cat("Please enter a valid number.\n")
    return()
  }

  cat("You selected:", chosen_number, "\n")

  if (chosen_number < 1 || chosen_number > 50) {
```

```
    cat("The number selected is beyond the range of 1 to 50.\n")
  } else if (chosen_number == 20) {
    cat("TRUE\n")
  } else {
    cat(chosen_number)
  }
}

exhaustive_search()
```

```
## Select a number between 1 and 50:
## Please enter a valid number.
```

```
## NULL
```

7. Change

a. Write a function that prints the minimum number of bills that must be paid, given the price of the snack. Input: Price of snack (a random number divisible by 50) Output: Minimum number of bills needed to purchase a snack.

```
min_bills <- function(price) {
  bills <- c(1000, 500, 200, 100, 50)
  count <- 0

  for (bill in bills) {
    num_bills <- price %/% bill
    count <- count + num_bills
    price <- price %% bill
  }

  return(count)
}

snack_price <- as.numeric(readline(prompt = "Enter the price of the snack (divisible by 50): "))
```

```
## Enter the price of the snack (divisible by 50):
```

```
if (!is.na(snack_price) && snack_price %% 50 == 0) {
  cat("Minimum number of bills needed:", min_bills(snack_price), "\n")
} else {
  cat("Price must be a number divisible by 50.\n")
}
```

```
## Price must be a number divisible by 50.
```

8. The following is each student's math score for one semester. Based on this, answer the following questions.

a. Create a dataframe from the above table. Write the R codes and its output.

```
name <- c("Annie","Thea","Steve","Hanna")
grade1 <- c(85, 65, 75, 95)
grade2 <- c(65, 75, 55, 75)
grade3 <- c(85, 90, 80, 100)
grade4 <- c(100, 90, 85, 90)

grades <- data.frame(Name = name, Grade1 = grade1, Grade2 = grade2, Grade4 = grade4)
```

```
grades
```

```
##    Name Grade1 Grade2 Grade4
## 1 Annie     85     65    100
## 2  Thea     65     75     90
## 3 Steve     75     55     85
## 4 Hanna     95     75     90
```

b. Without using the rowMean function, output the average score of students whose average math score over 90 points during the semester. write R code and its output.

```
average_scores <- (grades$Grade1 + grades$Grade2 + grades$Grade3 + grades$Grade4) / 4
students_above_90 <- grades[average_scores > 90, ]

if (nrow(students_above_90) > 0) {
    students_above_90_avg <- sum(average_scores[average_scores > 90]) / nrow(students_above_90)
} else {
    students_above_90_avg <- NA
}

students_above_90_avg
```

```
## [1] NA
```

c. Without using the mean function, output as follows for the tests in which the average score was less than 80 out of 4 tests.

```
name <- c("Annie", "Thea", "Steve", "Hanna")
grade1 <- c(85, 65, 75, 95)
grade2 <- c(65, 75, 55, 75)
grade3 <- c(85, 90, 80, 100)
grade4 <- c(100, 90, 85, 90)

grades <- data.frame(Name = name, Grade1 = grade1, Grade2 = grade2, Grade3 = grade3, Grade4 = grade4)

for (i in 1:nrow(grades)) {
  test_avg <- (grades[i, 2] + grades[i, 3] + grades[i, 4] + grades[i, 5]) / 4
  if (test_avg < 80) {
    cat("Name:", grades$Name[i], "- Average Score:", test_avg, "\n")
  }
}
```

```
## Name: Steve - Average Score: 73.75
```

d. Without using the max function, output as follows for students whose highest score for a semester exceeds 90 points.

```
for (i in 1:nrow(grades)) {
  highest_score <- grades[i, 2]
  for (j in 3:5) {
    if (grades[i, j] > highest_score) {
      highest_score <- grades[i, j]
    }
  }
  if (highest_score > 90) {
    cat("Name:", grades$Name[i], "- Highest Score:", highest_score, "\n")
  }
```

```
}
```

```
## Name: Annie - Highest Score: 100
## Name: Hanna - Highest Score: 100
```