CS544 Module 1 Assignment (Self Graded)

Part1) 50 points

The following sample data shows the scores of 10 students in an exam:

45, 80, 83, 78, 75, 77, 79, 83, 83, 100

Using R, assign the above data as a vector in the same order to a variable, say, *scores*. Do the following:

a) Using indexing, show the expression for accessing the first and last items. The code should not hardcode the value 10 for the number of items.
Sample output:

```
 [1]  45 100
```

b) Using comparison operators, write the expression for scores less than the mean, computed as mean(scores).
Sample output:

```
 [1]  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
```

c) Using logical indexing and the expression from b), return all the scores that are less than the mean.
Sample output:

```
 [1] 45 78 75 77
```

d) Using *rep* function, create a sequence, as the same length as scores, of alternating TRUE, FALSE values. Using this sequence, return every other element from the scores. The code should not hardcode the value 10 for the number of scores. You can assume that there are even number of values in scores.
Sample output:

```
 [1] 45 83 75 79 83
```

e) Using the paste function with LETTERS, show the code for the following output.
The code should not hardcode the value 10 for the number of scores.
Sample output:

```
 [1] "A=45"  "B=80"  "C=83"  "D=78"  "E=75"  "F=77"  "G=79"  "H=83"  "I=83"  "J=100"
```

f) Create a matrix of size 2 x 5 using the scores data. The first five values belong to the first row of the matrix. Assign the result to the variable, *scores.matrix*, and display the result.
Sample output:

```
      [,1]  [,2]  [,3]  [,4]  [,5]
[1,]   45   80   83   78   75
[2,]   77   79   83   83  100
```

g) Without hardcoding the value 5, show the code for displaying the first and last columns of the matrix.
Sample output:

```
      [,1] [,2]
[1,]   45   75
[2,]   77  100
```

h) Assign row names for the *scores.matrix* as Student_1, Student_2,… and column names as Quiz_1, Quiz_2 …. The code should not hard code the values 2 and 5.
Sample output:

	Quiz_1	Quiz_2	Quiz_3	Quiz_4	Quiz_5
Student_1	45	80	83	78	75
Student_2	77	79	83	83	100

Part 2) 50 points

Using the temperatures data from Mount Washington observatory (https://www.mountwashington.org/experience-the-weather/mount-washington-weather-archives/normals-means-and-extremes.aspx), create a data frame, say *weather.info*, using the column names: Month, Monthly_Average, Daily_Max_Average, Daily_Min_Average, Record_High, and Record_Low. Do not use the years for the last two columns. For the rows, do not use the *Annual* row. A snapshot of the resulting data frame, taking into account the above modifications, is shown below:

Month	Monthly_Avg	DailyMax_Avg	DailyMin_Avg	Record_High	Record_Low
January	4.7	13.6	-4.1	48	-47
February	6.1	14.7	-2.4	43	-46
March	12.8	20.7	5.0	54	-38
April	23.9	30.4	17.4	60	-20
May	35.5	41.3	29.8	66	-2
June	45.0	50.4	39.5	72	8
July	49.1	54.1	44.0	71	24
August	48.1	53.3	43.0	72	20
September	41.6	47.1	36.1	69	9
October	30.2	36.4	24.0	62	-5
November	20.7	28.1	13.3	52	-20
December	10.1	18.4	1.7	47	-46

a) Show the code for creating the data frame and display the resulting data frame. This is not an automated code. Create the values manually and then the data frame.

b) Show the summary for Monthly_Avg, DailyMax_Avg, DailyMin_Avg, Record_High, and Record_Low.

c) Show the data frame sliced using the columns Month, Record_High, and Record_Low.

d) Show the data frame sliced using the first and last row. Do not hard code 12 in the expression, i.e., the code should work for a data frame of any size.

e) Show all rows of the data frame whose DailyMax_Avg is greater than 50.

f) Modify the data by adding a new column, Record_Deviation, showing the difference between the Record_High and Record_Low. Display the new resulting data frame.