DAY 2 LAB PROGRAMS

1. 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.

PROGRAM

```
print("Two vectors of different lengths:")
v1 = c(1, 2, 3, 4)
v2 = c(11, 12, 13, 14, 15)
print(v1)
print(v2)
result = array(c(v1,v2),dim = c(3,3,2))
print("New array:")
print(result)
print("The second row of the second matrix of the array:")
print(result[2,,2])
print("The element in the 3rd row and 3rd column of the 1st matrix:")
print(result[3,3,1])
```

OUTPUT

```
[1] "Two vectors of different lengths:"
> v1 = c(1, 2, 3, 4)
> v2 = c(11, 12, 13, 14, 15)
> print(v1)
[1] 1 2 3 4
> print(v2)
[1] 11 12 13 14 15
> result = array(c(v1,v2),dim = c(3,3,2))
> print("New array:")
[1] "New array:"
> print(result)
, , 1

    [,1] [,2] [,3]
[1,] 1 4 13
[2,] 2 11 14
[3,] 3 12 15
, , 2

    [,1] [,2] [,3]
[1,] 1 4 13
[2,] 2 11 14
```

```
[3,] 3 12 15

> print("The second row of the second matrix of the array:")
[1] "The second row of the second matrix of the array:"
> print(result[2,,2])
[1] 2 11 14
> print("The element in the 3rd row and 3rd column of the 1st matrix:")
[1] "The element in the 3rd row and 3rd column of the 1st matrix:"
> print(result[3,3,1])
[1] 15
```

2. R program to combine three arrays so that the first row of the first array is followed by the first row of the second array and then the first row of the third array.

PROGRAM

```
num1 = rbind(rep("A",5), rep("B",5), rep("C",5))
print("num1")
print(num1)
num2 = rbind(rep("P",5), rep("Q",5), rep("R",5))
print("num2")
print(num2)
num3 = rbind(rep("X",5), rep("Y",5), rep("Z",5))
print("num3")
print(num3)
a = matrix(t(cbind(num1,num2,num3)),ncol=3, byrow=T)
print("Combine three arrays, taking one row from each one by one:")
print(a)
```

OUTPUT

```
[1] "num1"

    [,1] [,2] [,3] [,4] [,5]
[1,] "A" "A" "A" "A" "A"
[2,] "B" "B" "B" "B" "B"
[3,] "C" "C" "C" "C" "C"

[1] "num2"

    [,1] [,2] [,3] [,4] [,5]
[1,] "P" "P" "P" "P" "P"
[2,] "Q" "Q" "Q" "Q" "Q"
[3,] "R" "R" "R" "R" "R"
```

```
[1] "num3"

    [,1] [,2] [,3] [,4] [,5]
[1,] "X" "X" "X" "X" "X"
[2,] "Y" "Y" "Y" "Y" "Y"
[3,] "Z" "Z" "Z" "Z" "Z"

[1] "Combine three arrays, taking one row from each one by one:"

      [,1] [,2] [,3]
 [1,] "A" "A" "A"
 [2,] "A" "A" "P"
 [3,] "P" "P" "P"
 [4,] "P" "X" "X"
 [5,] "X" "X" "X"
 [6,] "B" "B" "B"
 [7,] "B" "B" "Q"
 [8,] "Q" "Q" "Q"
 [9,] "Q" "Y" "Y"
[10,] "Y" "Y" "Y"
[11,] "C" "C" "C"
[12,] "C" "C" "R"
[13,] "R" "R" "R"
[14,] "R" "Z" "Z"
[15,] "Z" "Z" "Z"
```

3. R program to create an array using four given columns, three given rows and two given tables and display the content of the array.

PROGRAM

```
array1 = array(1:15, dim=c(3,5,2))
print(array1)
```

OUTPUT

```
, , 1

    [,1] [,2] [,3] [,4] [,5]
[1,] 1 4 7 10 13
[2,] 2 5 8 11 14
[3,] 3 6 9 12 15

, , 2
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,] 1 4 7 10 13
[2,] 2 5 8 11 14
[3,] 3 6 9 12 15
```

4. R program to create a two-dimensional 5x3 array of sequence of even integers greater than 50.

PROGRAM

```
a <- array(seq(from = 50, length.out = 15, by = 2), c(5, 3))
print("Content of the array:")
print("5×3 array of sequence of even integers greater than 50:")
print(a)
```

OUTPUT

```
[1] "Content of the array:"

[1] "5×3 array of sequence of even integers greater than 50:"

     [,1] [,2] [,3]
[1,] 50 60 70
[2,] 52 62 72
[3,] 54 64 74
[4,] 56 66 76
[5,] 58 68 78
```

5. R program to extract 3 rd and 5 th rows with 1 st and 3 rd columns from a given data frame.

PROGRAM

```
exam_data = data.frame(
name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin',
'Jonas'),
score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
print("Extract 3rd and 5th rows with 1st and 3rd columns :")
result = exam_data[c(3,5),c(1,3)]
print(result)
```

OUTPUT

```
[1] "Original dataframe:

      name score attempts qualify
 1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
3 Katherine 16.5 2 yes
4 James 12.0 3 no
5 Emily 9.0 2 no
6 Michael 20.0 3 yes
7 Matthew 14.5 1 yes
8 Laura 13.5 1 no
9 Kevin 8.0 2 no
 10 Jonas 19.0 1 yes

[1] "Extract 3rd and 5th rows with 1st and 3rd columns :"

     name attempts
3 Katherine 2
5 Emily 2
```

6. R program to add a new column named country in a given data frame. Country<-c(“USA”, “USA”, “USA”, “USA”, “UK”, “USA”, “USA”, “India”, “USA”, “USA”)

   PROGRAM

```
exam_data = data.frame(
  name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura',
'Kevin', 'Jonas'),
  score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
print("New data frame after adding the 'country' column:")
exam_data$country =
c("USA","USA","USA","USA","USA","USA","USA","India","USA","USA")
print(exam_data)
```

   OUTPUT

```
[1] "Original dataframe:"
```

```
> print(exam_data)
      name score attempts qualify
 1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
3 Katherine 16.5 2 yes
4 James 12.0 3 no
5 Emily 9.0 2 no
6 Michael 20.0 3 yes
7 Matthew 14.5 1 yes
8 Laura 13.5 1 no
9 Kevin 8.0 2 no
 10 Jonas 19.0 1 yes
> print("New data frame after adding the 'country' column:")
[1] "New data frame after adding the 'country' column:"
> exam_data$country =
c("USA","USA","USA","USA","USA","USA","USA","India","USA","USA") >
print(exam_data)
      name score attempts qualify country
 1 Anastasia 12.5 1 yes USA
2 Dima 9.0 3 no USA
3 Katherine 16.5 2 yes USA
4 James 12.0 3 no USA
5 Emily 9.0 2 no USA
6 Michael 20.0 3 yes USA
7 Matthew 14.5 1 yes USA
8 Laura 13.5 1 no India
9 Kevin 8.0 2 no USA
 10 Jonas 19.0 1 yes USA
```

7. R program to add new row(s) to an existing data frame.

```
new_exam_data = data.frame(name = c(‘Robert’, ‘Sophia’),score = c(10.5, 9), attempts =
c(1, 3),qualify = c(‘yes’, ’no’))
```

PROGRAM

```
exam_data = data.frame(
  name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura',
'Kevin', 'Jonas'),
  score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
new_exam_data = data.frame(
 name = c('Robert', 'Sophia'),
  score = c(10.5, 9),
  attempts = c(1, 3),
  qualify = c('yes', 'no')
)
exam_data = rbind(exam_data, new_exam_data)
print("After adding new row(s) to an existing data frame:")
print(exam_data)
```

OUTPUT

```
[1] "Original dataframe:"

       name score attempts qualify
 1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
3 Katherine 16.5 2 yes
4 James 12.0 3 no
5 Emily 9.0 2 no
6 Michael 20.0 3 yes
7 Matthew 14.5 1 yes
8 Laura 13.5 1 no
9 Kevin 8.0 2 no
 10 Jonas 19.0 1 yes

[1] "After adding new row(s) to an existing data frame:"

       name score attempts qualify
 1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
3 Katherine 16.5 2 yes
4 James 12.0 3 no
5 Emily 9.0 2 no
6 Michael 20.0 3 yes
7 Matthew 14.5 1 yes
8 Laura 13.5 1 no
9 Kevin 8.0 2 no
 10 Jonas 19.0 1 yes
 11 Robert 10.5 1 yes
 12 Sophia 9.0 3 no
```

8. R program to sort a given data frame by name and score.

PROGRAM

```
exam_data = data.frame(
name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura',
'Kevin', 'Jonas'),
score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
print("dataframe after sorting 'name' and 'score' columns:")
exam_data = exam_data[with(exam_data, order(name, score)), ]
print(exam_data)
```

OUTPUT

```
[1] "Original dataframe:"

      name score attempts qualify
 1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
3 Katherine 16.5 2 yes
4 James 12.0 3 no
5 Emily 9.0 2 no
6 Michael 20.0 3 yes
7 Matthew 14.5 1 yes
8 Laura 13.5 1 no
9 Kevin 8.0 2 no
 10 Jonas 19.0 1 yes

[1] "dataframe after sorting 'name' and 'score' columns:"

      name score attempts qualify
 1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
5 Emily 9.0 2 no
4 James 12.0 3 no
 10 Jonas 19.0 1 yes
3 Katherine 16.5 2 yes
9 Kevin 8.0 2 no
8 Laura 13.5 1 no
```

```
7 Matthew 14.5 1 yes
6 Michael 20.0 3 yes
```

9. R program to save the information of a data frame in a file and display the information of the file.

PROGRAM

```
exam_data = data.frame(
  name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura',
'Kevin', 'Jonas'),
  score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
save(exam_data,file="data.student details")
load("data.student details")
file.info("data.student details")
```

OUTPUT

```
save(exam_data,file="data.student details")
> load("data.student details")
> file.info("data.student details")
                    size isdir mode mtime ctime
 data.student details 302 FALSE 666 2023-05-02 13:31:41 2023-05-02 13:27:41
                                          atime exe
data.student details 2023-05-02 13:31:41 no
```

10. R program to call the (built-in) dataset air quality. Check whether it is a data frame or not? Order the entire data frame by the first and second column. Remove the variables ‘Solar.R’ and ‘Wind’ and display the data frame.

PROGRAM

```
data = airquality
print("Original data: Daily air quality measurements in New York, May to September
 1973.")
print(class(data))
print(head(data,10))
result = data[order(data[,1]),]
 print("Order the entire data frame by the first and second column:")
print(result)
print(data)
data[,c("Solar.R")]=NULL
data[,c("Wind")]=NULL
 print("data.frame after removing 'Solar.R' and 'Wind' variables:")
print(data)
```

OUTPUT

```
print(data)
   Ozone Solar.R Wind Temp Month Day
 1 41 190 7.4 67 5 1
2 36 118 8.0 72 5 2
3 12 149 12.6 74 5 3 4 18 313
11.5 62 5 4 5 NA NA 14.3 56 5
5 6 28 NA 14.9 66 5 6 7 23 299
8.6 65 5 7
8 19 99 13.8 59 5 8
9 8 19 20.1 61 5 9
 10 NA 194 8.6 69 5 10 11 7 NA
 6.9 74 5 11 12 16 256 9.7 69 5
 12 13 11 290 9.2 66 5 13 14 14
 274 10.9 68 5 14 15 18 65 13.2
 58 5 15 16 14 334 11.5 64 5 16
 17 34 307 12.0 66 5 17 18 6 78
 18.4 57 5 18
 19 30 322 11.5 68 5 19 20 11 44
9.7 62 5 20 21 1 8 9.7 59 5 21
22 11 320 16.6 73 5 22 23 4 25
9.7 61 5 23
24 32 92 12.0 61 5 24 25 NA 66
16.6 57 5 25 26 NA 266 14.9 58
5 26 27 NA NA 8.0 57 5 27 28
23 13 12.0 67 5 28 29 45 252
14.9 81 5 29 30 115 223 5.7 79 5
30 31 37 279 7.4 76 5 31 32 NA
286 8.6 78 6 1 33 NA 287 9.7 74
6 2 34 NA 242 16.1 67 6 3 35
NA 186 9.2 84 6 4 36 NA 220
8.6 85 6 5 37 NA 264 14.3 79 6 6
38 29 127 9.7 82 6 7 39 NA 273
6.9 87 6 8
40 71 291 13.8 90 6 9 41 39 323
11.5 87 6 10 42 NA 259 10.9 93
6 11 43 NA 250 9.2 92 6 12 44
23 148 8.0 82 6 13 45 NA 332
13.8 80 6 14 46 NA 322 11.5 79
```

6 15 47 21 191 14.9 77 6 16 48
37 284 20.7 72 6 17 49 20 37 9.2
65 6 18 50 12 120 11.5 73 6 19
51 13 137 10.3 76 6 20 52 NA
150 6.3 77 6 21 53 NA 59 1.7 76
6 22 54 NA 91 4.6 76 6 23 55
NA 250 6.3 76 6 24 56 NA 135
8.0 75 6 25 57 NA 127 8.0 78 6
26 58 NA 47 10.3 73 6 27 59 NA
98 11.5 80 6 28 60 NA 31 14.9
77 6 29 61 NA 138 8.0 83 6 30
62 135 269 4.1 84 7 1 63 49 248
9.2 85 7 2 64 32 236 9.2 81 7 3
65 NA 101 10.9 84 7 4 66 64 175
4.6 83 7 5 67 40 314 10.9 83 7 6
68 77 276 5.1 88 7 7 69 97 267
6.3 92 7 8 70 97 272 5.7 92 7 9
71 85 175 7.4 89 7 10 72 NA 139
8.6 82 7 11 73 10 264 14.3 73 7
12 74 27 175 14.9 81 7 13 75 NA
291 14.9 91 7 14 76 7 48 14.3 80
7 15 77 48 260 6.9 81 7 16 78 35
274 10.3 82 7 17 79 61 285 6.3
84 7 18
80 79 187 5.1 87 7 19 81 63 220
11.5 85 7 20 82 16 7 6.9 74 7 21
83 NA 258 9.7 81 7 22 84 NA
295 11.5 82 7 23 85 80 294 8.6 86
7 24 86 108 223 8.0 85 7 25 87 20
81 8.6 82 7 26 88 52 82 12.0 86 7
27 89 82 213 7.4 88 7 28 90 50
275 7.4 86 7 29 91 64 253 7.4 83
7 30 92 59 254 9.2 81 7 31 93 39
83 6.9 81 8 1 94 9 24 13.8 81 8 2
95 16 77 7.4 82 8 3 96 78 NA 6.9
86 8 4 97 35 NA 7.4 85 8 5 98 66
NA 4.6 87 8 6 99 122 255 4.0 89
8 7 100 89 229 10.3 90 8 8 101
110 207 8.0 90 8 9 102 NA 222
8.6 92 8 10 103 NA 137 11.5 86 8
11 104 44 192 11.5 86 8 12 105
28 273 11.5 82 8 13 106 65 157
9.7 80 8 14 107 NA 64 11.5 79 8
15 108 22 71 10.3 77 8 16 109 59
51 6.3 79 8 17 110 23 115 7.4 76

```
8 18 111 31 244 10.9 78 8 19 112
44 190 10.3 78 8 20 113 21 259
15.5 77 8 21 114 9 36 14.3 72 8
22 115 NA 255 12.6 75 8 23 116
45 212 9.7 79 8 24 117 168 238
3.4 81 8 25 118 73 215 8.0 86 8
26 119 NA 153 5.7 88 8 27
 120 76 203 9.7 97 8 28
 121 118 225 2.3 94 8 29
 122 84 237 6.3 96 8 30
 123 85 188 6.3 94 8 31
 124 96 167 6.9 91 9 1
 125 78 197 5.1 92 9 2
 126 73 183 2.8 93 9 3
 127 91 189 4.6 93 9 4
 128 47 95 7.4 87 9 5
 129 32 92 15.5 84 9 6
 130 20 252 10.9 80 9 7
 131 23 220 10.3 78 9 8
 132 21 230 10.9 75 9 9
 133 24 259 9.7 73 9 10
 134 44 236 14.9 81 9 11
 135 21 259 15.5 76 9 12
 136 28 238 6.3 77 9 13
 137 9 24 10.9 71 9 14
 138 13 112 11.5 71 9 15
 139 46 237 6.9 78 9 16
 140 18 224 13.8 67 9 17
 141 13 27 10.3 76 9 18
 142 24 238 10.3 68 9 19
 143 16 201 8.0 82 9 20
 144 13 238 12.6 64 9 21
 145 23 14 9.2 71 9 22
 146 36 139 10.3 81 9 23
 147 7 49 10.3 69 9 24
 148 14 20 16.6 63 9 25
 149 30 193 6.9 70 9 26
 150 NA 145 13.2 77 9 27
 151 14 191 14.3 75 9 28
 152 18 131 8.0 76 9 29
 153 20 223 11.5 68 9 30
> data[,c("Solar.R")]=NULL
> data[,c("Wind")]=NULL
> print("data.frame after removing 'Solar.R' and 'Wind' variables:")
```

```
[1] "data.frame after removing 'Solar.R' and 'Wind' variables:" >
print(data)
   Ozone Temp Month Day
 1 41 67 5 1 2 36 72
5 2 3 12 74 5 3 4 18
62 5 4 5 NA 56 5 5
6 28 66 5 6 7 23 65
5 7 8 19 59 5 8 9 8
61 5 9
 10 NA 69 5 10 11 7
 74 5 11 12 16 69 5
 12 13 11 66 5 13 14
 14 68 5 14 15 18 58
 5 15 16 14 64 5 16
 17 34 66 5 17
 18 6 57 5 18 19 30
68 5 19 20 11 62 5
20 21 1 59 5 21 22
11 73 5 22 23 4 61 5
23 24 32 61 5 24
25  NA  57  5  25  26
NA  58  5  26  27  NA
57 5 27 28 23 67 5 28
29 45 81 5 29 30 115
79  5  30  31  37  76  5
31  32  NA  78  6  1  33
NA 74 6 2 34 NA 67
6 3 35 NA 84 6 4 36
NA 85 6 5 37 NA 79
6 6 38 29 82 6 7
39 NA 87 6 8 40 71
90 6 9
41 39 87 6 10 42 NA
93 6 11 43 NA 92 6
12 44 23 82 6 13 45
NA 80 6 14 46 NA
79 6 15 47 21 77 6 16
48 37 72 6 17 49 20
65 6 18 50 12 73 6 19
51 13 76 6 20 52 NA
77 6 21 53 NA 76 6
22 54 NA 76 6 23 55
NA 76 6 24 56 NA
75 6 25 57 NA 78 6
26 58 NA 73 6 27 59
```

NA 80 6 28 60 NA
77 6 29 61 NA 83 6
30 62 135 84 7 1 63
49 85 7 2 64 32 81 7
3 65 NA 84 7 4 66 64
83 7 5 67 40 83 7 6
68 77 88 7 7 69 97 92
7 8 70 97 92 7 9 71
85 89 7 10 72 NA 82
7 11 73 10 73 7 12 74
27 81 7 13 75 NA 91
7 14 76 7 80 7 15 77
48 81 7 16 78 35 82 7
17 79 61 84 7 18 80
79 87 7 19
81 63 85 7 20 82 16
74 7 21 83 NA 81 7
22 84 NA 82 7 23 85
80 86 7 24 86 108 85
7 25 87 20 82 7 26 88
52 86 7 27 89 82 88 7
28 90 50 86 7 29 91
64 83 7 30 92 59 81 7
31 93 39 81 8 1
94 9 81 8 2 95 16 82
8 3 96 78 86 8 4 97
35 85 8 5 98 66 87 8
6 99 122 89 8 7 100
89 90 8 8
101 110 90 8 9 102
NA 92 8 10 103 NA
86 8 11 104 44 86 8
12 105 28 82 8 13
106 65 80 8 14 107
NA 79 8 15 108 22 77
8 16 109 59 79 8 17
110 23 76 8 18 111 31
78 8 19 112 44 78 8
20 113 21 77 8 21
114 9 72 8 22 115 NA
75 8 23 116 45 79 8
24 117 168 81 8 25
118 73 86 8 26 119
NA 88 8 27 120 76 97
8 28

```
121 118 94 8 29
122 84 96 8 30
123 85 94 8 31
124 96 91 9 1
125 78 92 9 2
126 73 93 9 3
127 91 93 9 4
128 47 87 9 5
129 32 84 9 6
130 20 80 9 7
131 23 78 9 8
132 21 75 9 9
133 24 73 9 10
134 44 81 9 11
135 21 76 9 12
136 28 77 9 13
137 9 71 9 14
138 13 71 9 15
139 46 78 9 16
140 18 67 9 17
141 13 76 9 18
142 24 68 9 19
143 16 82 9 20
144 13 64 9 21
145 23 71 9 22
146 36 81 9 23
147 7 69 9 24
148 14 63 9 25
149 30 70 9 26
150 NA 77 9 27
151 14 75 9 28
152 18 76 9 29
153 20 68 9 30
```

11. R program to create a factor corresponding to height of women data set , which inbuilt in R, contains height and weights for a sample of women.

PROGRAM

```
data = women
print("Women data set of height and weights:")
print(data)
height_f = cut(women$height,7)
print("Factor corresponding to height:")
print(table(height_f))
```

OUTPUT

```
[1] "Women data set of height and weights:"

  height weight
 1 58 115
2 59 117
3 60 120
4 61 123
5 62 126
6 63 129
7 64 132
8 65 135
9 66 139
10 67 142
11 68 146
12 69 150
13 70 154
14 71 159
15 72 164

[1] "Factor corresponding to height:"

height_f
(58,60] (60,62] (62,64] (64,66] (66,68] (68,70] (70,72]
    3 2 2 2 2 2 2
```

12. R program to extract the five of the levels of factor created from a random sample from the LETTERS.(Part of the base R distribution)

PROGRAM

```
L = sample(LETTERS,size=100,replace=TRUE)
print("Original data:")
print(L)
f = factor(L)
print("Original factors:")
print(f)
print("Only five of the levels")
print(table(L[1:5]))
```

OUTPUT

```
L = sample(LETTERS,size=100,replace=TRUE)
> print("Original data:")
[1] "Original data:"
> print(L)
   [1] "L" "L" "W" "B" "Z" "S" "W" "L" "B" "P" "M" "A" "B" "B" "S" "U" "K" "U"
"M"
 [20] "W" "A" "C" "V" "O" "H" "I" "U" "B" "O" "T" "H" "F" "P" "T" "U" "X" "P" "F"
[39] "O" "C" "V" "T" "S" "Y" "Q" "S" "O" "D" "E" "T" "G" "X" "G" "N" "V" "V" "W"
 [58] "A" "D" "J" "B" "C" "L" "P" "U" "S" "B" "Y" "K" "P" "A" "M" "S" "E" "Q" "E"
 [77] "H" "O" "F" "C" "J" "P" "I" "J" "D" "K" "K" "B" "E" "U" "X" "X" "G" "V" "Q"
 [96] "M" "B" "F" "J" "Y"
> f = factor(L)
> print("Original factors:")
[1] "Original factors:"
> print(f)
  [1] L L W B Z S W L B P M A B B S U K U M W A C V O H I U B O T H F P T U
X P F O
 [40] C V T S Y Q S O D E T G X G N V V W A D J B C L P U S B Y K P A M S E
Q E H O
 [79] F C J P I J D K K B E U X X G V Q M B F J Y
 Levels: A B C D E F G H I J K L M N O P Q S T U V W X Y Z
> print("Only five of the levels")
[1] "Only five of the levels"
> print(table(L[1:5]))

B L W Z
 1 2 1 1
```