100 R Vector and Matrix Exercises

This is the R Version of 100 numpy exercises

Install and load package Magrittr (1 Star)

```
# install.packages("magrittr")
library(magrittr)
```

Print the R version (1 Star)

```
print(version$version.string)
```

```
## [1] "R version 3.3.0 (2016-05-03)"
```

Create a null vector of size 10 (1 Star)

```
rep(NA, 10) %>% print
```

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

How to get the examples of the add (+) function from the command line ? (1 Star)

```
example(`+`)
```

```
##
## +> x <- -1:12
##
## +> x + 1
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13
##
## +> 2 * x + 3
##  [1]  1  3  5  7  9 11 13 15 17 19 21 23 25 27
##
## +> x %% 2 #-- is periodic
##  [1] 1 0 1 0 1 0 1 0 1 0 1 0 1 0
##
## +> x %/% 5
##  [1] -1  0  0  0  0  0  1  1  1  1  1  2  2  2
```

Create a null vector of size 10 but the fifth value which is 1 (1 Star)

```
z <- rep(NA, 10)
z[5] <- 1
print(z)
```

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

Create a vector with values ranging from 10 to 49 (1 Star)

```
10:49 %>% print
```

```
##  [1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
## [24] 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
```

Reverse a vector (first element becomes last) (1 Star)

```
1:10 %>% rev %>% print
```

```
##  [1] 10  9  8  7  6  5  4  3  2  1
```

Create a 3x3 matrix with values ranging from 0 to 8 (1 Star)

```
matrix(0:8, ncol = 3, nrow = 3) %>% print
```

```
##      [,1] [,2] [,3]
## [1,]    0    3    6
## [2,]    1    4    7
## [3,]    2    5    8
```

Find indices of non-zero elements from (1,2,0,0,4,0) (1 Star)

```
c(1,2,0,0,4,0) %>% as.logical %>% which %>% print
```

```
## [1] 1 2 5
```

Create a 3x3 identity matrix (1 Star)

```
diag(3) %>% print
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

Create a 3x3x3 array with random values (1 Star)

```
runif(3^3) %>% array(c(3, 3, 3)) %>% print
```

```
## , , 1
##
##           [,1]      [,2]      [,3]
## [1,] 0.2837146 0.5623968 0.1740079
## [2,] 0.3874040 0.2231199 0.3093261
## [3,] 0.5550827 0.5212689 0.5532940
##
## , , 2
##
##             [,1]      [,2]      [,3]
```

```
## [1,] 0.005782446 0.7316805 0.7347370
## [2,] 0.075080524 0.2511427 0.8047678
## [3,] 0.180569264 0.3915239 0.6229788
##
## , , 3
##
##           [,1]      [,2]      [,3]
## [1,] 0.6766013 0.5497991 0.9069164
## [2,] 0.4454482 0.1120752 0.5620832
## [3,] 0.8687898 0.4990225 0.6482685
```

Create a 10x10 array with random values and find the minimum and maximum values (1 Star)

```
z <- array(runif(10^2), c(10,10))
min(z) %>% print
```

```
## [1] 0.009664419
```

```
max(z) %>% print
```

```
## [1] 0.997822
```

Create a random vector of size 30 and find the mean value (1 Star)

```
runif(30) %>% mean %>% print
```

```
## [1] 0.4552356
```

Create a 2d array with 1 on the border and 0 inside (1 Star)

```
z <- array(1, c(4, 5))
z[2:(nrow(z) - 1), 2:(ncol(z) - 1)] <- 0
print(z)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    1    1    1
## [2,]    1    0    0    0    1
## [3,]    1    0    0    0    1
## [4,]    1    1    1    1    1
```

What is the result of the following expression ? (1 Star)

```
0 * NaN
```

```
## [1] NaN
```

```
NA == NA
```

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

```
NA == NaN
```

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

```
Inf > NA
```

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

```
NaN - NaN
```

```
## [1] NaN
```

```
0.3 == 3 * 0.1
```

```
## [1] FALSE
```

Create a 5x5 matrix with values 1,2,3,4 just below the diagonal (1 Star)

```
z <- diag(0, ncol = 5, nrow = 5)
z[row(z) - 1 == col(z)] <- 1:4
print(z)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    1    0    0    0    0
## [3,]    0    2    0    0    0
## [4,]    0    0    3    0    0
## [5,]    0    0    0    4    0
```

Create a 8x8 matrix and fill it with a checkerboard pattern (1 Star)

```
z <- array(0, c(8, 8))
z[1:nrow(z) %% 2 != 0, 1:ncol(z) %% 2 == 0] <- 1
z[1:nrow(z) %% 2 == 0, 1:ncol(z) %% 2 != 0] <- 1
print(z)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    0    1    0    1    0    1    0    1
## [2,]    1    0    1    0    1    0    1    0
## [3,]    0    1    0    1    0    1    0    1
## [4,]    1    0    1    0    1    0    1    0
## [5,]    0    1    0    1    0    1    0    1
## [6,]    1    0    1    0    1    0    1    0
## [7,]    0    1    0    1    0    1    0    1
## [8,]    1    0    1    0    1    0    1    0
```

Consider a (6,7,8) shape array, what is the index (x,y,z) of the 100th element ?