Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh

TRUNG TÂM TIN HỌC

Lập trình R

Bài 7: *Matrix, Array và List*

Phòng LT & Mạng

https://csc.edu.vn/lap-trinh-va-csdl/R-Programming-Language-for-Data-Science 190

2020

Nội dung

1. **Matrix**
2. **Array**
3. **List**

Matrix

- Là đối tượng dữ liệu của R. Trong đó, các phần tử cùng kiểu dữ liệu được sắp xếp vào ma trận chữ nhật hai chiều.

- Ta thường sử dụng ma trận có các phần tử number để thực hiện tính toán.

Matrix

- Sử dụng matrix() để tạo ma trận, với data là dữ liệu vector, nrow là số dòng, ncol là số cột, byrow nếu = TRUE thì các phần tử đưa vào tạo matrix sẽ lần lượt theo dòng, dimnames là tên dòng và cột

Matrix

- Ví dụ

```
> # Elements are arranged sequentially by row.
> M <- matrix(c(1:12), nrow = 4, ncol = 3, byrow = TRUE)
> print(M)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
>
> # Elements are arranged sequentially by column.
> N <- matrix(c(1:12), nrow = 4, byrow = FALSE)
> print(N)
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
>
> N <- matrix(c(1:12), ncol = 4, byrow = TRUE)
> print(N)
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
>
> # Define the column and row names.
> rownames = c("row1", "row2", "row3", "row4")
> colnames = c("col1", "col2", "col3")
>
> P <- matrix(c(1:12), nrow = 4, byrow = TRUE, dimnames = list(rownames, colnames))
> print(P)
     col1 col2 col3
row1    1    2    3
row2    4    5    6
row3    7    8    9
row4   10   11   12
```


Matrix

❑ Truy xuất phần tử trong matrix

- Dùng matrix_name[row_index, column_index]

- Ví dụ:

	col1	col2	col3
row1	1	2	3
row2	4	5	6
row3	7	8	9
row4	10	11	12

```
> # Access the element at 3rd column and 1st row.
> print(P[1,3])
[1] 3
>
> # Access the element at 2nd column and 4th row.
> print(P[4,2])
[1] 11
>
> # Access only the  2nd row.
> print(P[2,])
col1 col2 col3
   4    5    6
>
> # Access only the 3rd column.
> print(P[,3])
row1 row2 row3 row4
   3    6    9   12
```

❑Tính toán trong matrix

- Có thể cộng, trừ, nhân, chia hai ma trận với nhau. Kết quả sau khi tính toán cũng là ma trận. Hai ma trận đem tính toán phải có cùng số dòng, số cột với nhau.
- Có thể dot product hai ma trận. Kết quả sau khi tính toán là ma trận. Số cột của ma trận 1 phải bằng số dòng của ma trận 2. (n,m) %*% (m,p) = (n,p)
- Có thể dot product ma trận và vector. Kết quả sau khi tính toán là vector. Số cột của ma trận phải bằng số phần tử của vector. (n,m) %*% (m) = (n)

- Ví dụ:

```
> M1 <- matrix(c(1, 3, 5, 7, 9, 11, 13, 15), nrow = 2, byrow = TRUE)
> print(M1)
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    9   11   13   15
> M2 <- matrix(c(0, 2, 4, 6, 8, 10, 12, 14), nrow = 2, byrow = TRUE)
> print(M2)
     [,1] [,2] [,3] [,4]
[1,]    0    2    4    6
[2,]    8   10   12   14
> # Add the matrices.
> result_add <- M1 + M2
> cat("Result of addition","\n")
Result of addition
> print(result_add)
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]   17   21   25   29
```

Matrix

- Ví dụ:

```
> # Subtract the matrices
> result_sub <- M1 - M2
> cat("Result of subtraction","\n")
Result of subtraction
> print(result_sub)
     [,1] [,2] [,3] [,4]
[1,]    1    1    1    1
[2,]    1    1    1    1
> # Multiply the matrices.
> result_mul <- M1 * M2
> cat("Result of multiplication","\n")
Result of multiplication
> print(result_mul)
     [,1] [,2] [,3] [,4]
[1,]    0    6   20   42
[2,]   72  110  156  210
> # Divide the matrices
> result_div <- M1 / M2
> cat("Result of division","\n")
Result of division
> print(result_div)
      [,1] [,2]     [,3]     [,4]
[1,]   Inf  1.5 1.250000 1.166667
[2,] 1.125  1.1 1.083333 1.071429
```


Matrix

- Ví dụ

```
mat1 <- matrix(c(1:12), ncol = 4)
mat2 <- matrix(c(1:8), nrow=4)
print(mat1)
print(mat2)

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

mat3 <- mat1 %*% mat2
print(mat3)

     [,1] [,2]
[1,]   70  158
[2,]   80  184
[3,]   90  210
```

Matrix

- Ví dụ

```
# matix
vec <- c(1:3)
mat <- matrix(c(1:9), ncol=3)
print(vec)
print(mat)
```

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

```
# dot product
vec_new <- mat %*% vec
print(vec_new)
```

```
     [,1]
[1,]   30
[2,]   36
[3,]   42
```


Nội dung

1. Matric
2. Array
3. List

Array

- Là đối tượng dữ liệu của R
- Dùng để lưu trữ dữ liệu nhiều chiều. Ví dụ: (2, 3, 4): 4 ma trận, mỗi ma trận chứa 2 dòng 3 cột.
- Array chỉ có thể lưu trữ data type

Array

- Sử dụng array() để tạo mảng, với vectors là dữ liệu và dim(rows, columns, matrics)
  - Ví dụ:

```
v1 <- c(1:6)
v2 <- c(7:15)
```

```
array1 <- array(v1, dim = c(3, 2, 2))
print(array1)

, , 1

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

, , 2

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

```
array2 <- array(c(v1, v2), dim = c(3, 4, 2))
print(array2)

, , 1

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

, , 2

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

Array

- **Đặt tên cho row và column: sử dụng dimnames = list(rownames, columnames, matrixnames)**
  - Ví dụ:

```
row_names = c("row1", "row2", "row3")
col_names = c("col1", "col2", "col3", "col4")
mat_names = c("matrix1", "matrix2")
array3 <- array(c(v1, v2), dim = c(3, 4, 2), dimnames = list(row_names, col_names, mat_names))
print(array3)

, , matrix1

     col1 col2 col3 col4
row1    1    4    7   10
row2    2    5    8   11
row3    3    6    9   12

, , matrix2

     col1 col2 col3 col4
row1   13    1    4    7
row2   14    2    5    8
row3   15    3    6    9
```


Array

- **Truy xuất phần tử trong array**
  - Dùng array_name[row_index, column_index, matrix_index]
  - Ví dụ:

```
, , matrix1

     col1 col2 col3 col4
row1    1    4    7   10
row2    2    5    8   11
row3    3    6    9   12

, , matrix2

     col1 col2 col3 col4
row1   13    1    4    7
row2   14    2    5    8
row3   15    3    6    9
```

```
print(array3[3, 2, 1])

[1] 6

print(array3[1,,2])

col1 col2 col3 col4
  13    1    4    7

print(array3[,3,1])

row1 row2 row3
   7    8    9
```

```
print(array3[,,2])

     col1 col2 col3 col4
row1   13    1    4    7
row2   14    2    5    8
row3   15    3    6    9
```

Array

- **Tìm sum/min/max/mean của các phần tử trong các dòng c(1) /cột c(2)/ matrix c(3) của một array qua tất cả các matrix**
  - Dùng apply(array, c(index), tên_function)
  - Ví dụ:

```
, , matrix1

     col1 col2 col3 col4
row1    1    4    7   10
row2    2    5    8   11
row3    3    6    9   12

, , matrix2

     col1 col2 col3 col4
row1   13    1    4    7
row2   14    2    5    8
row3   15    3    6    9
```

```
print(apply(array3, c(1), sum))

row1 row2 row3
  47   55   63

print(apply(array3, c(2), min))

col1 col2 col3 col4
   1    1    4    7

print(apply(array3, c(3), mean))

matrix1 matrix2
   6.50    7.25
```


Nội dung

1. Matrix
2. Array
3. List

List

- Là đối tượng dữ liệu của R, chứa danh sách các phần tử với các kiểu dữ liệu khác nhau như number, string, vector, matrix, list…

List

- Sử dụng list() để tạo danh sách
  - Ví dụ:

```
> # Create a list containing strings, numbers, vectors and a logical values.
> list_data <- list("hello", "R", 12, 25.5, c(21,32,11), TRUE)
> print(list_data)
[[1]]
[1] "hello"

[[2]]
[1] "R"

[[3]]
[1] 12

[[4]]
[1] 25.5

[[5]]
[1] 21 32 11

[[6]]
[1] TRUE
```

List

❑Đặt tên cho các phần tử trong list

- Các phần tử trong list có thể đặt tên để dễ dàng gọi sử dụng khi cần.
- Sử dụng: names(list_name) <- c(name_1, name_2, name_3,…)

List

- Ví dụ:

```
> # Create a list containing a vector, a matrix,list and logic value
> list_1 <- list(c("Red","Green","Yellow"), matrix(c(1:8), nrow = 2),
+                list("green",TRUE, 12.5), TRUE)
>
> # Give names to the elements in the list.
> names(list_1) <- c("Traffic Light", "A_Matrix", "A Inner list", "Pass")
> print(list_1)
$`Traffic Light`
[1] "Red"    "Green"  "Yellow"

$A_Matrix
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8

$`A Inner list`
$`A Inner list`[[1]]
[1] "green"

$`A Inner list`[[2]]
[1] TRUE

$`A Inner list`[[3]]
[1] 12.5


$Pass
[1] TRUE
```

List

❑ Truy xuất phần tử trong list

- Dùng chỉ số: list_name[index]
- Dùng tên: list_name$name_of_element

```
$`Traffic Light`
[1] "Red"    "Green"  "Yellow"

$A_Matrix
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8

$`A Inner list`
$`A Inner list`[[1]]
[1] "green"

$`A Inner list`[[2]]
[1] TRUE

$`A Inner list`[[3]]
[1] 12.5


$Pass
[1] TRUE
```

```
> # Access the first element of the list.
> print(list_1[1])
$`Traffic Light`
[1] "Red"    "Green"  "Yellow"

> # Access the list element using the name of the element.
> print(list_1$A_Matrix)
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
```


List

❑ Thao tác trên list

- Ta có thể thêm phần tử vào cuối list, xóa phần tử cuối list, cập nhật phần tử trong list.
- Ví dụ

```
> # Add element at the end of the list.
> list_1[4] <- "New element"
> print(list_1[4])
$Pass
[1] "New element"

>
> # Remove the last element.
> list_1[4] <- NULL
>
> # Print the 4th Element.
> print(list_1[4])
$<NA>
NULL

>
> # Update the 3rd Element.
> list_1[3] <-  TRUE
> print(list_1[3])
$`A Inner list`
[1] TRUE
```

❑Trộn list

- list_new <- c(list_1, list_2)
- Ví dụ:

```
> # Create two lists.
> list1 <- list(1,TRUE, c("Red", "Green", "Yellow"))
> list2 <- list(TRUE, "hello", 12.5)
>
> # Merge the two lists.
> merged.list <- c(list1,list2)
>
> # Print the merged list.
> print(merged.list)
[[1]]
[1] 1

[[2]]
[1] TRUE

[[3]]
[1] "Red"    "Green"  "Yellow"

[[4]]
[1] TRUE

[[5]]
[1] "hello"

[[6]]
[1] 12.5
```


❑Chuyển list thành vector

- Dùng unlist(list_name)
- Ví dụ:

```
> # Create lists.
> list1 <- list(1:5)
> list2 <-list(6:10)
> # Convert the lists to vectors.
> v1 <- unlist(list1)
> v2 <- unlist(list2)
>
> print(v1)
[1] 1 2 3 4 5
> print(v2)
[1]  6  7  8  9 10
>
> # Now add the vectors
> result <- v1+v2
> print(result)
[1]  7  9 11 13 15
```

Chapter 7: Matrix, Array, List

Exercise 1: Tạo matrix

- Tạo 3 vector x, y, z, mỗi vector có 3 phần tử số nguyên
- Tạo một matrix 3x3 từ 3 vector trên, sắp dữ liệu vào theo dòng
- Đặt tên cho từng dòng trong vector là r1, r2, r3
- Đặt tên cho từng cột trong vector là c1, c2, c3
- In matrix vừa tạo
- Kiểm tra xem dữ liệu vừa tạo có phải là matrix hay không?
- Tạo ma trận chuyển vị của ma trận vừa tạo

Exercise 2: Tạo và tính toán trên matrix

- Tạo ma trận matrix_b có 4x3, có giá trị là các số nguyên từ 1:12 xếp theo cột, với tên dòng là a, b, c, d và tên cột là x, y, z
- Tạo ma trận chuyển vị tB
- In ma trận
- In ma trận chuyển vị
- Tạo ma trận con subB từ ma trận B, ma trận 3x3 với dữ liệu là 3 dòng cuối của ma trận B.
- In ma trận con
- Tạo ma trận matrix_A có 3x3, có giá trị là các số nguyên từ 1:9 xếp theo cột, với tên dòng là a, b, c, d và tên cột là x, y, z
- Thực hiện các phép tính 3*A, A + subB, A - subB. In các kết quả tương ứng

Exercise 3: Tạo và tính toán trên matrix

- Tạo ma trận matrix A1 có nxn phần tử có giá trị ngẫu nhiên từ 1 -> nxn
- Tạo ma trận matrix A2 có nxm phần tử có giá trị ngẫu nhiên từ 1 -> nxm
- Giả sử biết: A1 * M = A2 => Hãy tính và in ra M
- In dòng 1 của M
- In cột 1 của M
- In giá trị của M tại vị trí dòng 3 cột 2 nếu n>=3 và m>=2

Exercise 4: Tạo và tính toán trên array

- Tạo một array A3 có 2 ma trận, mỗi ma trận có kích thước 3x3. Giá trị các phần tử từ 1:18 điền vào theo dòng với tên cho dòng là r1, r2 cho cột là c1, c2, c3, c4 và cho ma trận là matrix1, matrix2, matrix3.
- In array này
- In matrix2 của array A1
- In tổng theo dòng của tất cả các ma trận

- In tổng theo cột của tất cả các ma trận
- In tổng theo ma trận của array
- In tổng dòng 1 ma trận 1
- In tích cột 2, ma trận 2

Exercise 5: Tạo và tính toán trên array

- Tạo một array A4 có 2 ma trận, mỗi ma trận có kích thước 5x5. Mỗi phần tử trong ma trận có giá trị >=5, <=50 và chia hết cho 2.
- In array này
- Cho biết vị trí và giá trị của tất cả các phần tử chia hết cho cả 2, 3 và 6. Cho biết tổng của các phần tử này

Exercise 6: Tạo và tính toán trên list

- Tạo list x từ 2 vector p, q: p <- c(2,7,8), q <- c("A", "B", "C")
- In giá trị của x tại vị trí thứ 2
- In giá trị của phần tử thứ 2 trong element thứ 2 trong list
- Thay thế phần tử thứ 2 trong element thứ 2 trong list bằng "K"
- Cho vector t<-c(1, 3, 4), thêm vector này vào list
- In list sau khi thêm
- Cho list a như sau: a <- list ("x"=5, "y"=10, "z"=15)
- Tính tổng các phần tử trong list
- Cho list như sau:Newlist <- list(a=1:10, b="Good morning", c="Hi")
- Cộng thêm 1 vào các phần tử trong element a trong list

Exercise 7: Tạo và tính toán trên list

- Cho 3 vector: year <- c(2008:2018), month <- c(1:12), day <- c(1:31)
- Tạo list chứa 3 vector year, month, day trên
- Cập nhật year trong list thành c(2010:2020)
- Bỏ năm đầu tiên trong year của list
- Cho vector x: x <- c(1,3,4,7,11,18,29)
- Tạo list có 3 phần tử như sau: x*2, x/2 và sqrt(x)
- In list
- In 3 phần tử giữa trong element sqrt(x) của list
- Tạo list mới chứa 2 list trên
- Tạo list con chỉ chứa year, x*2, day từ list mới tạo ra trên
- Cho biết chiều dài của element x*2
- Cho biết giá trị của phần tử cuối cùng trong year

Exercise 8: Tạo và tính toán trên list

- Cho list: v <- list(c("a", "b", "c"), c(1,2,3),10)

- Đặt tên cho 3 phần tử trong list lần lượt là "one", "two", "three"
- Cho biết chiều dài của từng phần tử trong list
- Cho list: z <- list(c("a", "b", "c"), c(1,3,5,7),TRUE)
- Tạo list mới với các phần tử chỉ có trong y
- Tạo list mới với các phần tử chỉ có trong z

Gợi ý

Exercise 1: Tạo matrix

In [1]:
```
x <- c(1, 2, 3)
y <- c(4, 5, 6)
z <- c(7, 8, 9)

v1 <- c(x, y, z)
```

In [2]:
```
matrix_1 <-matrix(v1, nrow = 3, ncol = 3, byrow = TRUE,
                  dimnames = list(c("r1", "r2", "r3"), c("c1", "c2", "c3")))
print("Matrix 1")
print(matrix_1)

t_matrix_1 <- t(matrix_1)
print("Ma tran chuyen vi cua matrix_1")
print(t_matrix_1)
```

```
[1] "Matrix 1"
   c1 c2 c3
r1  1  2  3
r2  4  5  6
r3  7  8  9
[1] "Ma tran chuyen vi cua matrix_1"
   r1 r2 r3
c1  1  4  7
c2  2  5  8
c3  3  6  9
```

Exercise 2: Tạo và tính toán trên matrix

In [3]:
```
# tao ma tran 12 phan tu co gia tri tu 1:12
matrix_B <- matrix(data = c(1:12), nrow = 4,
                   dimnames = list(c("a", "b", "c", "d"), c("x", "y", "z")))
print("Matrix B:")
print(matrix_B)
tB <- t(matrix_B)
print("Transpose of Matrix B:")
print(tB)
subB<-matrix_B[2:dim(matrix_B)[1],1:3]
print("Sub of Matrix B:")
print(subB)
```

```
[1] "Matrix B:"
  x y  z
a 1 5  9
b 2 6 10
c 3 7 11
d 4 8 12
[1] "Transpose of Matrix B:"
  a  b  c  d
x 1  2  3  4
y 5  6  7  8
z 9 10 11 12
[1] "Sub of Matrix B:"
  x y  z
b 2 6 10
c 3 7 11
d 4 8 12
```

In [4]:
```
matrix_A <- matrix(data = c(1:9), nrow = 3,
                   dimnames = list(c("a", "b", "c"), c("x", "y", "z")))
print("Matrix A:")
print(matrix_A)

# 3A
A3 <- matrix_A * 3
print("3*A:")
print(A3)
A_plus_subB <- matrix_A + subB
print("A + sub B:")
print(A_plus_subB)

A_minus_subB <- matrix_A - subB
print("A - sub B:")
print(A_minus_subB)
```

```
[1] "Matrix A:"
  x y z
a 1 4 7
b 2 5 8
c 3 6 9
[1] "3*A:"
  x  y  z
a 3 12 21
b 6 15 24
c 9 18 27
[1] "A + sub B:"
  x  y  z
a 3 10 17
b 5 12 19
c 7 14 21
[1] "A - sub B:"
   x  y  z
a -1 -2 -3
b -1 -2 -3
c -1 -2 -3
```

Exercise 3: Tạo và tính toán trên matrix

In [5]:
```
n = as.integer(readline(prompt = "Nhap n: "))
```

```
Nhap n: 4
```

In [6]:
```
# tao A1 co nxn phan tu ngau nhien
A1 <- matrix(runif(n*n, min = 1, max = n*n), n, n)
print("A1")
print(A1)
```

```
[1] "A1"
          [,1]      [,2]      [,3]      [,4]
[1,]  2.277316  1.249831 13.206876 14.916963
[2,] 14.837797  8.513637  5.938895  6.172207
[3,]  5.067707 15.971319  6.741166  7.932833
[4,]  4.503694  9.343787  2.743459  8.318420
```

In [8]:
```
m = as.integer(readline(prompt = "Nhap m: "))
```

```
Nhap m: 3
```

In [9]:
```
# tao A2 co n*m phan tu ngau nhien
A2 <- matrix(runif(n*m, min = 1, max = n*m), n, m)
print("A2")
print(A2)
```

```
[1] "A2"
          [,1]     [,2]      [,3]
[1,]  6.623821 9.430260  7.443201
[2,] 10.139421 5.209143  7.641052
[3,]  9.724673 2.254542 10.181982
[4,]  9.270946 3.341724 11.076617
```

In [10]:
```
# tinh M biet A1 * M = A2
M <- solve(A1, A2)
print("Matrix M")
print(M)
```

```
[1] "Matrix M"
           [,1]       [,2]       [,3]
[1,]  0.3697695  0.2244360  0.1484121
[2,]  0.3050683 -0.2344358  0.3557611
[3,] -0.3772869  0.1332281 -0.7291828
[4,]  0.6960691  0.4996075  1.0920994
```

In [11]:
```
# hay in dong 1 cua M
print("Row 1")
print(M[1,])
```

```
[1] "Row 1"
[1] 0.3697695 0.2244360 0.1484121
```

In [12]:
```
# hay in cot 1 cua M
print("Column 1")
print(M[,1])
```

```
[1] "Column 1"
[1]  0.3697695  0.3050683 -0.3772869  0.6960691
```

In [13]:
```
# hay in gia tri dong 3 cot 2 cua M
if (n>=3 && m>=2){
    print("M[3][2]")
    print(M[3,2])
}
```

```
[1] "M[3][2]"
[1] 0.1332281
```

Exercise 4: Tạo và tính toán trên array

In [14]:
```
# tao array A3
matrix_1 <- matrix(c(1:9),nrow = 3, byrow = TRUE)
matrix_2 <- matrix(c(10:18),nrow = 3, byrow = TRUE)

A3 <- array(c(matrix_1, matrix_2), dim = c(3, 3, 2),
            dimnames = list(c("r1", "r2", "r3"),
                            c("c1", "c2", "c3"),
                            c("matrix1", "matrix2")))
print("Array A3:")
print(A3)
```

```
[1] "Array A3:"
, , matrix1

   c1 c2 c3
r1  1  2  3
r2  4  5  6
r3  7  8  9

, , matrix2

   c1 c2 c3
r1 10 11 12
r2 13 14 15
r3 16 17 18
```

In [15]:
```
#tong theo dong cua tat ca matrix
print("Tong theo dong cua tat ca matrix")
print(apply(A3, c(1),sum))
#tong theo cot cua tat ca matrix
print("Tong theo cot cua tat ca matrix")
print(apply(A3, c(2),sum))
# tong tat ca matrix
print("Tong theo matrix cua array")
print(apply(A3, c(3),sum))

print("Tong dong 1, matrix1")
print(sum(A3[1,,1]))
print("Tich cot 2, matrix2")
print(prod(A3[, 2,2]))
```

```
[1] "Tong theo dong cua tat ca matrix"
r1 r2 r3
39 57 75
[1] "Tong theo cot cua tat ca matrix"
c1 c2 c3
51 57 63
[1] "Tong theo matrix cua array"
matrix1 matrix2
     45     126
[1] "Tong dong 1, matrix1"
[1] 6
[1] "Tich cot 2, matrix2"
[1] 2618
```

Exercise 5: Tạo và tính toán trên array

In [16]:
```
v <- c()
i <-1
while(i<=50){
  x = floor(runif(1, min = 5, max = 50))
  if (x%%2==0){
    v[i] <-x
    i <- i + 1
  }
}

print(v)
```

```
 [1] 14 22 46 18 14 46 26 24 24 24 36  8  6 32 26 44 10 34 30 18 46 16 26 14
[26] 20 40 18 48 24 36 16 44 26 10 38 24 16 28 44 34 16 14 36  8 34 46 36 32
```

In [18]:
```
A4 <- array(v, dim=c(5, 5, 2))
print("Array A4:")
print(A4)
```

```
[1] "Array A4:"
, , 1

     [,1] [,2] [,3] [,4] [,5]
[1,]   14   46   36   44   46
[2,]   22   26    8   10   16
[3,]   46   24    6   34   26
[4,]   18   24   32   30   14
[5,]   14   24   26   18   24

, , 2

     [,1] [,2] [,3] [,4] [,5]
[1,]   20   36   38   34   34
[2,]   40   16   24   16   46
[3,]   18   44   16   14   36
[4,]   48   26   28   36   32
[5,]   24   10   44    8   34
```

In [19]:
```
rows <- nrow(A4)
print(rows)

cols <- ncol(A4)
print(cols)

dims <- dim(A4)[3]
print(dims)
```

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

```
In [20]: i <- 1
         j <- 1
         k <- 1
         tong <-0
         while(i<=dims){
           #print(paste("Matrix ", i))
           j <-1
           while(j<=rows){
             #print(paste("row ", j))
             k <-1
             while(k<=cols){
               #print(paste("col ", k))
               if(A4[j,k,i]%% 2 == 0 & A4[j,k,i]%% 3 == 0 & A4[j,k,i]%% 6 == 0){
                 print(paste("A4[", j,",", k,",", i,"] =", A4[j,k,i]))
                 tong <- tong + A4[j,k,i]
               }
               k <- k+1
             }
             j <- j+1
           }
           i <- i+1
         }
         print(paste("Tong =", tong))

         [1] "A4[ 1 , 3 , 1 ] = 36"
         [1] "A4[ 3 , 2 , 1 ] = 24"
         [1] "A4[ 3 , 3 , 1 ] = 6"
         [1] "A4[ 4 , 1 , 1 ] = 18"
         [1] "A4[ 4 , 2 , 1 ] = 24"
         [1] "A4[ 4 , 4 , 1 ] = 30"
         [1] "A4[ 5 , 2 , 1 ] = 24"
         [1] "A4[ 5 , 4 , 1 ] = 18"
         [1] "A4[ 5 , 5 , 1 ] = 24"
         [1] "A4[ 1 , 2 , 2 ] = 36"
         [1] "A4[ 2 , 3 , 2 ] = 24"
         [1] "A4[ 3 , 1 , 2 ] = 18"
         [1] "A4[ 3 , 5 , 2 ] = 36"
         [1] "A4[ 4 , 1 , 2 ] = 48"
         [1] "A4[ 4 , 4 , 2 ] = 36"
         [1] "A4[ 5 , 1 , 2 ] = 24"
         [1] "Tong = 426"
```

Exercise 6: Tạo và tính toán trên list

In [21]:
```
p <- c(2,7,8)
q <- c("A", "B", "C")
x <- list(p, q)
print("List x:")
print(x)
```

```
[1] "List x:"
[[1]]
[1] 2 7 8

[[2]]
[1] "A" "B" "C"
```

In [22]:
```
# in gia tri cua x tai vi tri thu 2
print(paste("list[2]: ",toString(x[2])))
# in gia tri phan tu thu 2 cua vector thu 2 trong list
print(paste("vector[2] of list[2]:", x[[2]][2]))
# thay the phan tu thu 2 cua vector thu 2 trong list thanh "K"
x[[2]][2]<- "K"
print("List after changing")
print(x)
```

```
[1] "list[2]:  c(\"A\", \"B\", \"C\")"
[1] "vector[2] of list[2]: B"
[1] "List after changing"
[[1]]
[1] 2 7 8

[[2]]
[1] "A" "K" "C"
```

In [23]:
```
# them t<-c(1, 3, 4) vao list
#cach 1
t<- c(1, 3, 4)
x <- c(x, list(t))
print("List after insert:")
print(x)
#cach 2
x[length(x)+1]<-list(t)
print(x)
```

```
[1] "List after insert:"
[[1]]
[1] 2 7 8

[[2]]
[1] "A" "K" "C"

[[3]]
[1] 1 3 4

[[1]]
[1] 2 7 8

[[2]]
[1] "A" "K" "C"

[[3]]
[1] 1 3 4

[[4]]
[1] 1 3 4
```

In [24]:
```
a <- list ("x"=5, "y"=10, "z"=15)
print("List a:")
print(a)
#tinh tong cua list
print(paste("Sum of list:", sum(unlist(a))))
```

```
[1] "List a:"
$x
[1] 5

$y
[1] 10

$z
[1] 15

[1] "Sum of list: 30"
```

In [25]:
```
Newlist <- list(a=1:10, b="Good morning", c="Hi")
print("Newlist:")
print(Newlist)
#cong them 1 vao cac phan tu cua element a trong list
Newlist$a <-Newlist$a+1
print(Newlist)
```

```
[1] "Newlist:"
$a
 [1]  1  2  3  4  5  6  7  8  9 10

$b
[1] "Good morning"

$c
[1] "Hi"

$a
 [1]  2  3  4  5  6  7  8  9 10 11

$b
[1] "Good morning"

$c
[1] "Hi"
```

Exercise 7: Tạo và tính toán trên list

In [26]:
```
# cho 3 vector day, month, year => tao 1 list chua 3 element nay
year <- c(2008:2018)
month <- c(1:12)
day <- c(1:31)

# list
year_month_day.list <- list("year" = year, "month" = month, "day" = day)
print("List year_month_day:")
print(year_month_day.list)
```

```
[1] "List year_month_day:"
$year
 [1] 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018

$month
 [1]  1  2  3  4  5  6  7  8  9 10 11 12

$day
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30 31
```

In [27]:
```
# cap nhat year thanh c(2010:2020)
year_month_day.list$year <- c(2010:2020)
print("List year_month_day after changing:")
print(year_month_day.list)
```

```
[1] "List year_month_day after changing:"
$year
 [1] 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

$month
 [1]  1  2  3  4  5  6  7  8  9 10 11 12

$day
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 2
[26] 26 27 28 29 30 31
```

In [28]:
```
# bo nam dau tien trong year cua list
#x <- length(year_month_day.list$year)
year_month_day.list$year <- year_month_day.list$year[-1]
print("After drop the first year in year:")
print(year_month_day.list)
```

```
[1] "After drop the first year in year:"
$year
 [1] 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

$month
 [1]  1  2  3  4  5  6  7  8  9 10 11 12

$day
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
[26] 26 27 28 29 30 31
```

In [29]:
```
x <- c(1,3,4,7,11,18,29)
cal.list <- list("x*2"= x*2,"x/2"=x/2, "sqrt(x)"=sqrt(x))
print("List calculator:")
print(cal.list)
#tra ve cac phan tu 3-5 $sqrt(x)
print("3 phan tu giua")
print(cal.list$`sqrt(x)`[3:5])
```

```
[1] "List calculator:"
$`x*2`
[1]  2  6  8 14 22 36 58

$`x/2`
[1]  0.5  1.5  2.0  3.5  5.5  9.0 14.5

$`sqrt(x)`
[1] 1.000000 1.732051 2.000000 2.645751 3.316625 4.242641 5.385165

[1] "3 phan tu giua"
[1] 2.000000 2.645751 3.316625
```

In [30]:
```
#tao 1 list moi co 2 list tren
year_month_day_cal.list <- c(year_month_day.list, cal.list)
print("year_month_day_cal.list: ")
print(year_month_day_cal.list)
```

```
[1] "year_month_day_cal.list: "
$year
 [1] 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

$month
 [1]  1  2  3  4  5  6  7  8  9 10 11 12

$day
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30 31

$`x*2`
[1]  2  6  8 14 22 36 58

$`x/2`
[1]  0.5  1.5  2.0  3.5  5.5  9.0 14.5

$`sqrt(x)`
[1] 1.000000 1.732051 2.000000 2.645751 3.316625 4.242641 5.385165
```

In [32]:
```
# tao mot list con chua year, x*2 va day tu list moi tao ra tren
year_x2_day <- year_month_day_cal.list[c("year", "x*2", "day")]
print("list year_x2_day:")
print(year_x2_day)
```

```
[1] "list year_x2_day:"
$year
 [1] 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

$`x*2`
[1]  2  6  8 14 22 36 58

$day
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
[26] 26 27 28 29 30 31
```

In [33]:
```
# chieu dai cua x*2
print(paste("length of x*2:", length(year_x2_day$`x*2`)))
print(paste("last value of year:",
            year_x2_day$year[length(year_x2_day$year)]))
```

```
[1] "length of x*2: 7"
[1] "last value of year: 2020"
```

Exercise 8: Tạo và tính toán trên list

In [34]:
```
# cho list
y <- list(c("a", "b", "c"), c(1,2,3),10)
#dat ten
names(y) <- c("one", "two", "three")
print("List y:")
print(y)
```

```
[1] "List y:"
$one
[1] "a" "b" "c"

$two
[1] 1 2 3

$three
[1] 10
```

In [35]:
```
# chieu dai
print(paste("Length of one, two, three are:",
            length(y$one), ",",
            length(y$two), ",",
            length(y$three)))
```

```
[1] "Length of one, two, three are: 3 , 3 , 1"
```

In [36]:
```
# z
z <- list(c("a", "b", "c"), c(1,3,5,7),TRUE)
print("list z:")
print(z)
```

```
[1] "list z:"
[[1]]
[1] "a" "b" "c"

[[2]]
[1] 1 3 5 7

[[3]]
[1] TRUE
```

In [37]:
```
# in y not in z
t <- setdiff(y, z)
print("in y not in z:")
print(t)
#in z not in y
f <- setdiff(z, y)
print("in z not in y")
print(f)
```

```
[1] "in y not in z:"
[[1]]
[1] 1 2 3

[[2]]
[1] 10

[1] "in z not in y"
[[1]]
[1] 1 3 5 7

[[2]]
[1] TRUE
```