第三次上机实验

生信 2001 张子栋 2020317210101

GitHub 地址: [MarkdownNotes/R at main · Bluuur/MarkdownNotes (github.com)](github.com)

1. 安装 R 包 `qrcode` , 并用函数 `qrcode_gen` 生成对应自己学号的二维码

```
1  > install.packages("qrcode")
2  > library("qrcode")
3  > qrcode_gen("2020317210101")
```


2. 编写一个求 3 个向量交集的函数

```
1  > myIntersect <- function(v1, v2, v3){+
2  + for (val1 in v1) {
3  +     for (val2 in v2) {
4  +         for (val3 in v3) {
5  +             if (val1==val2 && val2==val3) {
6  +                 print(val1)
7  +             }
8  +         }
9  +     }
10 + }
11 + }
12 > v1=c(1,2,3,4,5)
13 > v2=c(2,3)
14 > v3=c(1,2,3,4,5,6,7)
15 > myIntersect(v1,v2,v3)
16 [1] 2
17 [1] 3
```

3. 矩阵操作: 生成如下 15×15 的字符类型矩阵 A

$$A = \begin{pmatrix} a_1 & b_2 & a_3 & b_4 & \cdots & a_{15} \\ c & b_2 & a_3 & b_4 & \cdots & a_{15} \\ c & c & a_3 & b_4 & \cdots & a_{15} \\ c & c & c & b_4 & \cdots & a_{15} \\ \vdots & \vdots & \vdots & \vdots & \ddots & a_{15} \\ c & c & c & c & c & a_{15} \end{pmatrix}$$

提示: `paste` 函数生成字母和数字的组合

```
> myMatrix <- function() {
+     a <- c()
+     for (i in 1:15) {
+         for (j in 1:i) {
+             if ((i %% 2) == 1) {
+                 a <- c(a, paste("a", i, sep = ""))
+             } else {
+                 a <- c(a, paste("b", i, sep = ""))
+             }
+         }
+         for (k in 1:(15 - i)) {
+             a <- c(a, "c")
+         }
+     }
+     return(matrix(a, ncol = 15, nrow = 15, byrow = F))
+ }
> myMatrix()
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
 [1,] "a1" "b2" "a3" "b4" "a5" "b6" "a7" "b8" "a9" "b10" "a11" "b12" "a13" "b14" "a15"
 [2,] "c"  "b2" "a3" "b4" "a5" "b6" "a7" "b8" "a9" "b10" "a11" "b12" "a13" "b14" "a15"
 [3,] "c"  "c"  "a3" "b4" "a5" "b6" "a7" "b8" "a9" "b10" "a11" "b12" "a13" "b14" "a15"
 [4,] "c"  "c"  "c"  "b4" "a5" "b6" "a7" "b8" "a9" "b10" "a11" "b12" "a13" "b14" "a15"
 [5,] "c"  "c"  "c"  "c"  "a5" "b6" "a7" "b8" "a9" "b10" "a11" "b12" "a13" "b14" "a15"
 [6,] "c"  "c"  "c"  "c"  "c"  "b6" "a7" "b8" "a9" "b10" "a11" "b12" "a13" "b14" "a15"
 [7,] "c"  "c"  "c"  "c"  "c"  "c"  "a7" "b8" "a9" "b10" "a11" "b12" "a13" "b14" "a15"
 [8,] "c"  "c"  "c"  "c"  "c"  "c"  "c"  "b8" "a9" "b10" "a11" "b12" "a13" "b14" "a15"
 [9,] "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "a9" "b10" "a11" "b12" "a13" "b14" "a15"
[10,] "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "b10" "a11" "b12" "a13" "b14" "a15"
[11,] "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"   "a11" "b12" "a13" "b14" "a15"
[12,] "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"   "c"   "b12" "a13" "b14" "a15"
[13,] "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"   "c"   "c"   "a13" "b14" "a15"
[14,] "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"  "c"   "c"   "c"   "c"   "b14" "a15"
```

```
[15,] "c"   "c"   "c"   "c"   "c"   "c"   "c"   "c"   "c"   "c"   "c"   "c"   "c"
      "c"   "a15"
```