

## 第七周——分支与循环

### 题目目的

- (一) 掌握分支语句的应用。
- (二) 掌握循环语句的使用。
- (三) 掌握使用分支语句和循环语句解决简单问题。
- (四) 掌握next语句和break语句的应用。

### 题目

#### 题目一：if语句与if-else语句的应用。打开脚本文件test0701.R，完成下面操作。

- (1) 若n是偶数，则用print函数显示“n是偶数！”，否则不显示。
- (2) 若m是偶数，则用print函数显示“m是偶数！”，否则显示“m是奇数！”。

```
n = rpois(1,75)
m = rpois(1,80)
```

#### 题目二：多路分支的应用。打开脚本文件test0702.R，完成下面操作。

假设score表示分数，根据score判断等级，90分及以上为优秀，80分至89分为良好，70分至79分为中等，60分至69分为及格，60分以下为不及格。

```
score = round(runif(1,40,100), digits = 0)
```

#### 题目三：分支语句的应用。打开脚本文件test0703.R，用分支语句完成下面操作。

- (1) 以a, b, c为边长，判断是否可以构成一个三角形。
- (2) x, y, z是三个任意的整数，用分支语句找出这三个数中的最大值，并用cat函数显示x, y, z三个数及其最大值。

```
x = rpois(1, 60)
y = rpois(1, 60)
z = rpois(1, 60)
a = rpois(1, 30)
b = rpois(1, 50)
c = rpois(1, 70)
```

#### 题目四：分支函数的应用。打开脚本文件test0704.R，完成下面操作。

- (1) 用ifelse函数产生一个新向量over40，向量的元素由“是”和“否”构成。产生规则为：对应于age中的元素，若大于40，则为“是”，否则为“否”。
- (2) 用ifelse函数找出x, y, z中的最大值，并用cat函数显示x, y, z三个数及其最大值。

```
age = round(runif(35, 18, 60))
x = rpois(1, 60)
y = rpois(1, 60)
z = rpois(1, 60)
```

#### 题目五：循环语句的应用。打开脚本文件test0705.R，用循环语句完成下面操作。

- (1) 用for循环计算并输出向量x的所有元素之和，并用sum函数验证计算结果的正确性。
- (2) 用while循环计算并输出向量x的所有元素之和。
- (3) 用repeat-break循环计算并输出向量x的所有元素之和。

```
x = round(runif(100, 1, 99),digits = 0)
```

#### 题目六：循环语句与switch函数的应用。打开脚本文件test0706.R，完成下面操作。

依据向量z的内容，用循环语句修改数据框deg列，修改规则为：若z中的元素为1，则deg列对应位置修改为“本科”；若为2，则修改为“硕士”；若为3，则修改为“博士”。

```
x = sprintf('%04d', 1:36)
y = round(runif(36, 22, 55), digits = 0)
z = rpois(36,75) %% 3 + 1
df = data.frame(ID = x, age = y, deg = z)
```

#### 题目七：next语句和break语句的应用。打开脚本文件test0707.R，完成下面操作。

- (1) 计算向量x中大于0的元素之和，且遇到大于30的元素时，立即停止计算，要求使用next语句和break语句控制程序流程。

(2) 用print函数显示计算结果（注：参考结果为696.87）。

```
set.seed(1)
x = round(rnorm(100, 10, 10), digits = 2)
```

题目八：循环语句的应用。打开脚本文件test0708.R，完成下面操作。

(1) 用循环计算计算向量x和向量y的相关系数，其定义如下：

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

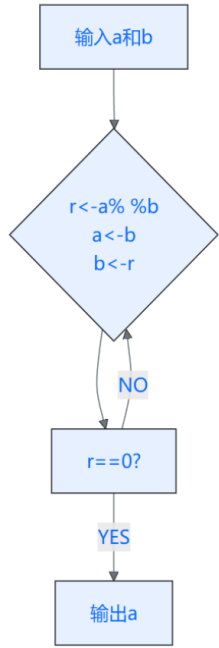
其中， $x_i$ 为向量x中的元素， $y_i$ 为向量y中的元素， $\bar{x}$ 和 $\bar{y}$ 为向量x和y的均值，可用mean函数计算，公式中求和运算（ $\sum$ ）要求用循环语句实现。

(2) 用print函数显示结果，并与cor(x, y)的返回结果对比。

```
x = rnorm(100, 5, 3)
y = runif(100, 0, 10)
cor(x,y)

[1] 0.0803554
```

题目九：算法实验一。打开脚本文件test0709.R，实现欧几里德算法求两个正整数的最大公约数，其中欧几里德算法的流程如下图所示。



```
a = rpois(1, 10000)
b = rpois(1, 500)
cat(a, '\t', b, '\n')
```

10013      509

题目十：算法实验二。打开脚本文件test0710.R，使用冒泡算法对向量x进行排序。

(1) 算法简要描述 x = c(5, 6, 3, 4, 1, 2)，由小到大排序，排序过程与体育课由矮到高排队相似。

- 第1趟：第6个与第5个比较，小的移到第5的位置，再与第4个比较，...，最小的那个（数字1）就冒泡到第1的位置。
- 第2趟：剩下的5个数（第2个到第6个），重复上述过程，这样第2小的数就排到了正确位置。
- 第3趟：剩下的4个数（第3个到第6个），重复上述过程，这样第3小的数排到了正确位置。
- 第4趟：剩下的3个数（第4个到第6个），重复上述过程，这样第4小的数排到了正确位置。
- 第5趟：剩下的2个数（第5个和第6个），重复上述过程，这样第5小的数排到了正确位置。
- 最后只剩下1个数了，就不用再比较了。

(2) 总结

- n个数字只需要n-1趟循环。
- 每趟都从最后那个开始，与前面的比较，若后面的小，则交换。因此，每趟需要做循环，且第k趟循环的最后那次比较，是第k+1个元素与第k个元素比较。

伪代码

```
n <- x的长度
k <- 从1到(n-1)
j <- 从n到(k+1)
if(x[j]<x[j-1]){交换x[j]与x[j-1]}
下一个j
下一个k
...
```

## 答案及解析

### 题目一：

```
n = rpois(1,75)
m = rpois(1,80)
if (n %%2 != 1){
  print('n是偶数! ')
}
```

[1] "n是偶数! "

```
if (m %%2 != 1){
  print('m是偶数!')
}else{
  print('m是奇数! ')
}
```

[1] "m是奇数! "

### 题目二：

```
score = round(runif(1,40,100), digits = 0)

if (score >= 90){
  print('优秀')
}else if(score >= 80){
  print('良好')
}else if(score >= 70){
  print('中等')
}else if(score >= 60){
  print('及格')
}else{
  print('不及格')
}
```

[1] "中等"

### 题目三：

```
x = rpois(1, 60)
y = rpois(1, 60)
z = rpois(1, 60)
a = rpois(1, 30)
b = rpois(1, 50)
c = rpois(1, 70)

if (a + b > c && a + c > b && b + c > a){
  print('可以构成三角形')
}else{
  print('不能构成三角形')
}
```

[1] "可以构成三角形"

```
max = 0
if (x>y){
  if(x>z){
    max <- x
  }else{
    max <- z
  }
}else{
  if(y>z){
    max <- y
  }else{
    max <- z
  }
}
cat(x,y,z,max)
```

68 41 53 68

### 题目四：

```
age = round(runif(35, 18, 60))
x = rpois(1, 60)
y = rpois(1, 60)
z = rpois(1, 60)

over40 <- ifelse(age > 40,"是",'否')
print(over40)
```

[1] "是" "是" "是" "是" "是" "否" "否" "否" "否" "否" "否" "否" "否" "是" "否"
[16] "是" "是" "否" "否" "否" "是" "否" "是" "是" "是" "是" "否" "是" "是" "否"
[31] "否" "否" "是" "是" "是"

```
tmax = ifelse(
  x > y,
  ifelse(x>z,x,z),
  ifelse(y>z,y,z)
)
cat(x,',',y,',',',z,'的最大值为',tmax)
```

51 , 70 , 54 的最大值为 70

题目五:

```
x = round(runif(100, 1, 99),digits = 0)
add <- 0
for (i in x){
  add <- add +i
}
ifelse(sum(x)==add,print('正确'),print('错误'))
```

[1] "正确"

[1] "正确"

```
i <- 1
while (i <= length(x)) {
  add <- add + x[i]
  i<- i +1
}

repeat{
  add <- add + x[i]
  i <- i + 1
  if (i > length(x)){
    break
  }
}
```

题目六:

```
x = sprintf('%04d', 1:36)
y = round(runif(36, 22, 55), digits = 0)
z = rpois(36,75) %% 3 + 1
df = data.frame(ID = x, age = y, deg = z)

for (i in 1:length(z)) {
  df$deg[i] <- switch(z[i],
                     '本科',
                     '硕士',
                     '博士')
}
df
```

	ID	age	deg
1	0001	44	博士
2	0002	47	硕士
3	0003	39	博士
4	0004	49	硕士
5	0005	39	硕士
6	0006	39	博士
7	0007	36	硕士
8	0008	34	硕士
9	0009	26	本科
10	0010	32	硕士
11	0011	31	本科
12	0012	47	博士
13	0013	48	硕士
14	0014	27	硕士
15	0015	39	本科
16	0016	42	博士
17	0017	39	硕士
18	0018	35	博士
19	0019	36	本科
20	0020	22	博士
21	0021	52	博士
22	0022	25	硕士
23	0023	39	本科
24	0024	49	本科
25	0025	42	本科
26	0026	36	硕士
27	0027	40	博士
28	0028	48	硕士
29	0029	28	硕士
30	0030	54	博士
31	0031	38	硕士
32	0032	53	博士
33	0033	52	硕士
34	0034	47	硕士
35	0035	44	博士
36	0036	43	本科

题目七:

```
set.seed(1)
x = round(rnorm(100, 10, 10),digits = 2)

s=0
x
```

[1]	3.74	11.84	1.64	25.95	13.30	1.80	14.87	17.38	15.76	6.95
[11]	25.12	13.90	3.79	-12.15	21.25	9.55	9.84	19.44	18.21	15.94
[21]	19.19	17.82	10.75	-9.89	16.20	9.44	8.44	-4.71	5.22	14.18
[31]	23.59	8.97	13.88	9.46	-3.77	5.85	6.06	9.41	21.00	17.63
[41]	8.35	7.47	16.97	15.57	3.11	2.93	13.65	17.69	8.88	18.81
[51]	13.98	3.88	13.41	-1.29	24.33	29.80	6.33	-0.44	15.70	8.65
[61]	34.02	9.61	16.90	10.28	2.57	11.89	-8.05	24.66	11.53	31.73
[71]	14.76	2.90	16.11	0.66	-2.54	12.91	5.57	10.01	10.74	4.10
[81]	4.31	8.65	21.78	-5.24	15.94	13.33	20.63	6.96	13.70	12.67
[91]	4.57	22.08	21.60	17.00	25.87	15.58	-2.77	4.27	-2.25	5.27

```
for(k in x){
  if(k<0)next
  if(k>30)break
  s = s+k
}
print(s)
```

[1] 696.87

## 题目八：

```
x = rnorm(100, 5, 3)
y = runif(100, 0, 10)
cor(x,y)
```

[1] 0.0803554

```
ymean <- mean(y)
xmean <- mean(x)

csum = 0
xsum = 0
ysum = 0
for (k in 1:length(x)) {
  csum = csum + (x[k]-xmean)*(y[k]-ymean)
  xsum = xsum + (x[k]-xmean)^2
  ysum = ysum + (y[k]-ymean)^2
}
pho = csum/sqrt(xsum*ysum)
print(pho)
```

[1] 0.0803554

## 题目九：

```
# 欧几里德算法求最大公约数
euclidean_gcd <- function(a, b) {
  # 确保输入为正整数
  a <- as.integer(a)
  b <- as.integer(b)

  # 处理可能的非正整数情况
  if (a <= 0 || b <= 0) {
    stop("输入必须是正整数")
  }

  # 欧几里德算法主循环
  repeat {
    r <- a %% b # 计算余数
    a <- b      # 更新a为原来的b
    b <- r      # 更新b为余数r

    # 当余数为0时，当前a即为最大公约数
    if (r == 0) {
      break
    }
  }

  return(a)
}

# 生成随机正整数
set.seed(123) # 设置随机种子，保证结果可复现
a <- rpois(1, 10000)
b <- rpois(1, 500)

# 确保生成的是正整数（如果随机生成0则重新生成）
while (a <= 0) {
  a <- rpois(1, 10000)
}
while (b <= 0) {
  b <- rpois(1, 500)
}

# 输出原始数据
cat("两个随机正整数为：", a, "\t", b, "\n")
```

两个随机正整数为： 9943 526

```
# 计算并输出最大公约数
gcd_result <- euclidean_gcd(a, b)
cat("它们的最大公约数是：", gcd_result, "\n")
```

它们的最大公约数是： 1

## 题目十：

```
# 冒泡排序算法实现
bubble_sort <- function(x) {
  n <- length(x) # 获取向量长度

  # 外层循环：n个元素需要n-1趟排序
  for (k in 1:(n-1)) {
    # 内层循环：从最后一个元素开始，到第k+1个元素结束
    for (j in n: (k+1)) {
      # 如果后面的元素小于前面的元素，则交换位置
      if (x[j] < x[j-1]) {
        # 交换元素
        temp <- x[j]
        x[j] <- x[j-1]
```

```
        x[j-1] <- temp
      }
    }
  }

  return(x) # 返回排序后的向量
}

# 测试向量
x <- c(5, 6, 3, 4, 1, 2)
cat("原始向量: ", x, "\n")
```

原始向量: 5 6 3 4 1 2

```
# 调用冒泡排序函数
sorted_x <- bubble_sort(x)
cat("排序后向量: ", sorted_x, "\n")
```

排序后向量: 1 2 3 4 5 6