

Основы программирования в R

Алла Тамбовцева

Практикум 3: функции, условия, цикл for

Задача 1

Напишите код, который запрашивает у пользователя его имя и фамилию и если оно совпадает Вашим, выводит на экран “Hello, [name][surname], welcome to R! I missed you”, если не совпадает, то выводит на экран “Sorry, R is not available now”. Подразумевается, что R знает Ваше имя (при желании можете написать функцию, которая вытаскивает Ваше имя пользователя на компьютере и сравнивает с ним).

```
try_to_start <- function(){
input <- readline(prompt = "Enter your name and surname: ")
line <- unlist(strsplit(input, split = " "))
user_name <- line[1]
user_surname <- line[2]

my_name <- "Alla"
my_surname <- "Tambovtseva"

if (my_name == user_name & my_surname == user_surname){
  print(sprintf("Hello, %s %s, welcome to R! I missed you.", user_name, user_surname))
} else{
  print("Sorry, R is not available now")
}
}
```

Более радикальный вариант: R сравнивает имя с именем пользователя из рабочей папки и если имена не совпадают, прекращает сессию RStudio:

```
username <- readline(prompt = "Enter your name: ")
path <- getwd()
myname <- unlist(strsplit(path, "/"))[3]

if (myname == username){
  print("Hello, you are welcome!")
} else{
  print("R is not available now.")
  quit()
}
```

Задача 2

Напишите код, который сохраняет число, которое ввел пользователь с клавиатуры (предполагается, что пользователь вводит только числа, причем в правильном формате — в качестве разделителя ис-

пользует точку), в переменную `x` и если `x` является целым числом, то выводит на экран сообщение "It is an integer.", а если `x` не является таковым, то выводит "It is not an integer".

```
x <- readline("Enter a number: ")
x
if (grepl("\\.", x)){
  n <- unlist(strsplit(x, split = "\\."))[2]
  if (n == "0"){
    print("It is an integer.")
  } else {
    print("It is not an integer.")
  }
} else{
  print("It is an integer.")
}
```

Вариант без работы с текстом:

```
x <- readline("Enter a number: ")
nn <- as.numeric(x)
if (nn %% 1 == 0){
  print("It is an integer.")
} else {
  print("It is not an integer.")
}
```

Задача 3

Напишите код, который запрашивает у пользователя число элементов вектора ("Enter number of elements:"), сохраняет его в переменную `n` и создает вектор заданной длины, состоящий из пропущенных значений.

Задача 4

Напишите код, который запрашивает у пользователя число элементов вектора ("Enter number of elements:"), сохраняет его в переменную `n` и создает вектор заданной длины, состоящий из пропущенных значений. Далее, если индекс элемента четный, то этот элемент заменяется на 1, если нечетный - на 0.

```
n <- as.integer(readline(prompt = "Enter number of elements: "))
v <- rep(NA, n)
for (i in 1:n){
  if (i %% 2 == 0){
    v[i] <- 1
  } else {
    v[i] <- 0
  }
}
v
```

Вариант с функцией:

```
gen_vect <- function(){
  n <- as.integer(readline(prompt = "Enter number of elements: "))
  v <- rep(NA, n)
```

```

for (i in 1:n){
  if (i %% 2 == 0){
    v[i] <- 1
  } else {
    v[i] <- 0
  }
}
v # return(n)
}
gen_vect()

```

Задача 5

Напишите код, который запрашивает у пользователя размерность матрицы (предполагается, что пользователь вводит число строк и столбцов через пробел), сохраняет их и создает единичную матрицу заданной размерности. Если невозможно создать единичную матрицу заданной размерности, на экран выводится сообщение “Impossible to create an identity matrix with such dimensions.”.

Подсказка: единичная матрица - квадратная матрица (число строк равно числу столбцов), на главной диагонали которой стоят 1, а все остальные элементы равны 0. Пример:

```

[1 0 0]
[0 1 0]
[0 0 1]

```

```

di <- as.integer(unlist(strsplit(readline("Enter dimensions: "), split = " ")))
rows <- di[1]
cols <- di[2]
if (rows != cols){
  print("Impossible to create an identity matrix with such dimensions.")
} else {
  M <- matrix(0, nrow = rows, ncol = cols)
  for (i in 1:rows){
    for (j in 1:cols){
      if (i == j){
        M[i, j] <- 1
      }
    }
  }
}
M

```

Без цикла, с функцией `diag()`:

```

diag(M)

di <- as.integer(unlist(strsplit(readline("Enter dimensions: "), split = " ")))
rows <- di[1]
cols <- di[2]
if (rows != cols){
  print("Impossible to create an identity matrix with such dimensions.")
} else {
  M <- matrix(0, nrow = rows, ncol = cols)
  diag(M) <- 1
}

```

```
}  
M
```

Задача 6

Напишите функцию, которая принимает на вход числовой вектор и возвращает вектор, состоящий из квадратов элементов вектора, поданного на вход. Можете для определенности назвать функцию `get_squares()`.

```
get_squares <- function(w){  
  ws <- w**2  
  return(ws) # ws  
}  
get_squares(c(4, -1, 0, 1))
```

Задача 7

Напишите функцию `smart_squares()`, которая принимает на вход вектор, и если он числовой, то возвращает вектор из квадратов его элементов, а если нет — возвращает вектор из пропущенных значений и выводит на экран сообщение “Your vector is not numeric.”

```
smart_squares <- function(w){  
  if (is.numeric(w)){  
    res <- w ** 2  
  } else {  
    res <- rep(NA, length(w))  
    print("Your vector is not numeric.")  
  }  
  return(res)  
}  
  
smart_squares(c(-1, 2, 3))  
smart_squares(c(-1, "2", 3))
```