

## Часть 1: Ввод и вывод

### Задача 1:

Напишите код, который запрашивает у пользователя его имя с клавиатуры и выводит на экран сообщение вида "Hello, name!".

#### Работа программы:

```
> task_1_1()
Введите имя: Bloodies
hello, Bloodies!
> |
```

#### Листинг:

```
task_1_1 <- function()
{
  name <- readline("Введите имя: ")

  cat(sprintf("Hello, %s!", name))
}
```

### Задача 2:

Напишите код, который запрашивает у пользователя два дробных числа (одно число — один запрос) и выводит на экран их сумму. Решите эту задачу в предположении, что:

- пользователь вводит дробные числа с десятичным разделителем в виде точки;
- пользователь вводит дробные числа с десятичным разделителем в виде запятой.

#### Работа программы:

```
> task_1_2()
Введите дробное число: 1,23
Введите дробное число: 5.3
Сумма чисел: 6.53
> |
```

С проверкой на неправильный ввод

```
> task_1_2()
Введите дробное число: фыв
Введите дробное число: а
Введите дробное число: 1,1
Введите дробное число: 2.2
Сумма чисел: 3.3Предупреждения:
1: В method(input) : в результате преобразования созданы NA
2: В method(input) : в результате преобразования созданы NA
>
```

#### Листинг:

```
task_1_2 <- function()
{
  num1 <- suppressWarnings(validate(as.numeric, "numeric"))
  num2 <- suppressWarnings(validate(as.numeric, "numeric"))

  cat("Сумма чисел:", num1 + num2);
}
```

### Задача 3:

Напишите код, который запрашивает у пользователя значение скорости в километрах в час и выводит на экран значение скорости в метрах в секунду.

## Работа программы:

```
> task_1_3()
Введите скорость целым числом в км/ч: 100
Скорость в метрах в секунду: 27.77778 м/с
>
```

## Листинг:

```
task_1_3 <- function()
{
  spd <- validate(as.integer, "speed")

  cat("Преобразованная скорость:", (spd*1000)/3600, "м/с")
}
```

## Часть 2: Векторы

### Задача 1:

Дан вектор `g`, в котором хранятся следующие значения:

```
1, 0, 2, 3, 6, 8, 12, 15, 0, NA, NA, 9, 4, 16, 2, 0
```

Создайте вектор `g`. Используя R, выведите на экран:

- первый элемент вектора;
- последний элемент вектора;
- элементы вектора с третьего по пятый включительно;
- элементы вектора, которые равны 2;
- элементы вектора, которые больше 4;
- элементы вектора, которые кратны 3 (делятся на 3 без остатка);
- элементы вектора, которые больше 4 и кратны 3;
- элементы вектора, которые или меньше 1, или больше 5;
- индексы элементов, которые равны 0;
- индексы элементов, которые не меньше 2 и не больше 8.

## Работа программы:

```
> task_2_1()
Первый элемент:      1
Последний элемент:  0
Элементы с 3 по 5:   2 3 6
Элементы равные 2:   2 NA NA 2
Элементы больше 4:   6 8 12 15 NA NA 9 16
Элементы кратные 3:   0 2 6 8 12 0 NA NA 4 16 2 0
Элементы кратные 3 и больше 4: 6 8 12 NA NA 16
Элементы меньше 1 или больше 5: 0 6 8 12 15 0 NA NA 9 16 0
Индексы равных 0:     2 9 16
Индексы не меньше 2 и больше 8: 3 4 5 13 15
>
```

## Листинг:

```
task_2_1 <- function()
{
  vector <- c(1, 0, 2, 3, 6, 8, 12, 15, 0, NA, NA, 9, 4, 16, 2, 0)

  cat('Первый элемент:      ', vector[1], '\n')
  cat('Последний элемент:   ', vector[length(vector)], '\n')
  cat('Элементы с 3 по 5:     ', vector[3:5], '\n')
  cat('Элементы равные 2:     ', vector[vector == 2], '\n')
```

```

cat('Элементы больше 4:      ', vector[vector > 4], '\n')
cat('Элементы кратные 3:      ', vector[vector %% 2 == 0], '\n')
cat('Элементы кратные 3 и больше 4: ', vector[vector > 4 & vector %% 2 == 0], '\n')
cat('Элементы меньше 1 или больше 5:', vector[vector < 1 | vector > 5], '\n')
cat('Индексы равных 0:          ', which(vector == 0), '\n')
cat('Индексы не меньше 2 и больше 8:', which(vector >= 2 & vector < 8), '\n')
}

```

## Задача 2:

Напишите код, который заменяет последний элемент любого вектора на пропущенное значение (NA). Ваш код должен работать для вектора любой длины.

### Работа программы:

```

> task_2_2()
Вектор с замененным последним элементом на NA: 1 200 0 TRUE 5 6 7 NA 2 -1 20 NA
> |

```

### Листинг:

```

task_2_2 <- function()
{
  vector <- c(1, 200, 0, TRUE, 5, 6, 7, NA, 2, -1, 20.0, "TEST")
  vector[length(vector)] = NA

  cat("Вектор с замененным последним элементом на NA:", vector)
}

```

## Задача 3:

Напишите код, который выводит на экран индексы пропущенных значений в векторе.

### Работа программы:

```

> task_2_3()
Индексы пропущенных элементов: 3 4 7 9
> |

```

### Листинг:

```

task_2_3 <- function()
{
  vector <- c(1, 200, NA, NA, TRUE, 5, NA, 7, NA, 2, -1, 20.0, "TEST")

  cat("Индексы пропущенных элементов:", which(is.na(vector)))
}

```

## Задача 4:

Напишите код, который считает, сколько пропущенных значений в векторе.

### Работа программы:

```

> task_2_4()
Число пропущенных элементов: 4
> |

```

### Листинг:

```

task_2_4 <- function()
{
  vector <- c(1, 200, NA, NA, TRUE, 5, NA, 7, NA, 2, -1, 20.0, "TEST")

```

```
cat("Число пропущенных элементов:", length(vector[is.na(vector)]))
}
```

### Задача 5:

Напишите код, который позволяет создать вектор из id (уникальных номеров) респондентов, если известно, что в опросе участвовало 100 респондентов.

#### Работа программы:

```
> task_2_5()
Вектор со 100 уникальными значениями: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 1
8 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 4
5 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 7
2 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 9
9 100
> |
```

#### Листинг:

```
task_2_5 <- function()
{
  amount = 100;
  vector <- c()
  for (c in 1:amount)
    vector[c] = c

  cat("Вектор со", amount, "уникальными значениями:", vector)
}
```

### Задача 6:

*Решение задачи предполагает рациональное решение с помощью функций R, не нужно создавать вектора перечислением большого числа элементов вручную.*

Известно, что в таблице хранятся показатели по 3 странам за 5 лет (см. ниже).

Создайте вектор который мог бы послужить столбцом `country` в этой таблице. Создайте вектор, который мог бы послужить столбцом `year` в этой таблице.

Таблица выглядит так:

	country	year
1	France	2000
2	France	2001
3	France	2002
4	France	2003
5	France	2004
6	Italy	2000
7	Italy	2001
8	Italy	2002
9	Italy	2003
10	Italy	2004
11	Spain	2000
12	Spain	2001
13	Spain	2002
14	Spain	2003
15	Spain	2004

### Работа программы:

```
> task_2_6()
  Country Year
1  France 2000
2  France 2001
3  France 2002
4  France 2003
5  France 2004
6   Italy 2000
7   Italy 2001
8   Italy 2002
9   Italy 2003
10  Italy 2004
11 Spain 2000
12 Spain 2001
13 Spain 2002
14 Spain 2003
15 Spain 2004
>
```

### Листинг:

```
task_2_6 <- function()
{
  repeats = 5;    # Количество повторений
  yearFrom = 2000; # Год начала отсчета
  countries = c()
  for (i in c("France", "Italy", "Spain"))
    countries <- c(countries, rep(i, repeats))
  years = c();
  for (i in 1:repeats)
    years[i] = yearFrom + i - 1

  table <- data.frame("Страна" = countries, "Год" = years)
  print(table)
}
```

### Задача 7:

Исследователь решил создать вектор `income`, в котором сохранены доходы нескольких респондентов:

```
income <- c(10000, 32000, 28000, 150000, 65000, 1573)
```

Исследователю нужно получить вектор `income_class`, состоящий из 0 и 1: 0 ставится, если доход респондента ниже среднего дохода, а 1 — если больше или равен среднему доходу. Создайте вектор `income_class`.

*Подсказка:* сначала можно посчитать среднее значение по вектору `income` и сохранить его в какую-нибудь переменную. Пользоваться встроенной функцией `mean()` нельзя.

### Работа программы:

```
> task_2_7()
вектор income_class: 0 0 0 1 1 0
>
```

### Листинг:

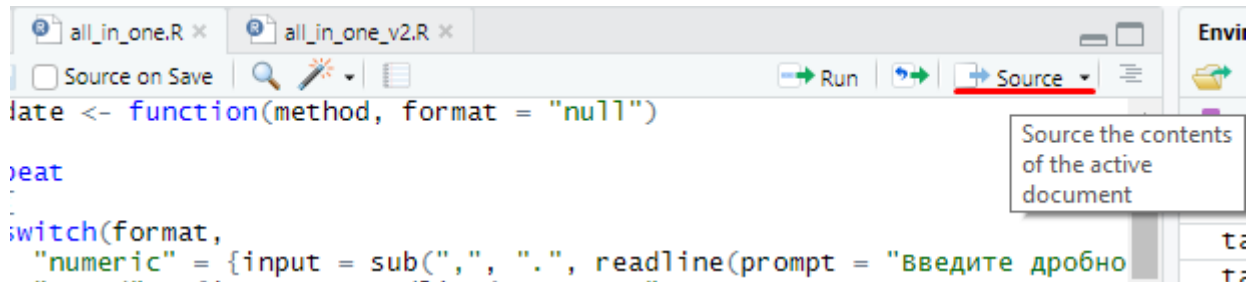
```
{
  income <- c(10000, 32000, 28000, 150000, 65000, 1573)
  average = sum(income) / length(income)

  income_class <- replace(temp <- replace(income, income < average, 0), temp >= average, 1)
```

```
cat("Вектор income_class:", income_class)
}
```

## Способы запуска:

- 1) запуск всего кода, чтобы инициализировать функции



- 2) запуск заданий:  
task\_[1]\_[2]()  
[1] – номер части  
[2] – номер задания из части

