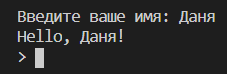
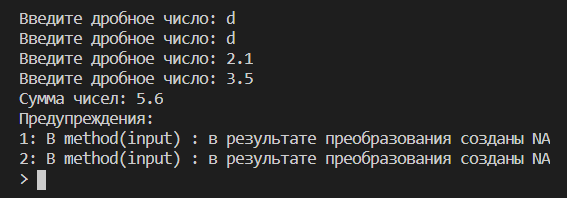
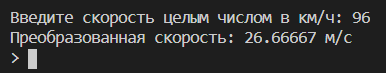
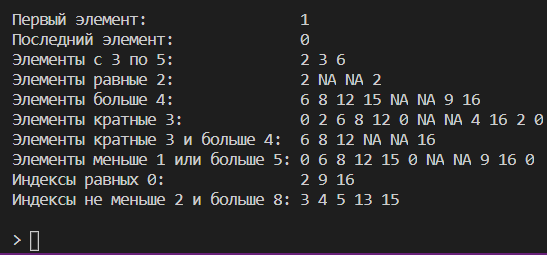
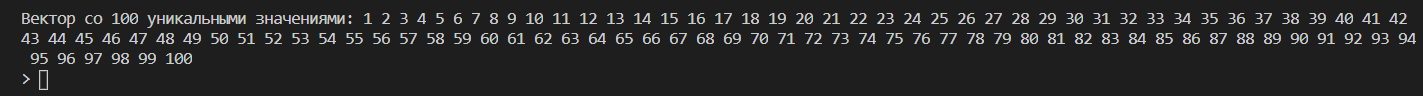
Результаты выполнения:

1. Ввод и вывод
   1. 
   2. 
   3. 
2. Векторы
   1. 
   2. 
   3. 
   4. 
   5. 
   6. 
   7. 

Код программы:

# ================================

#           Ввод и вывод

# ================================

# Вспомогательные методы

validate <- function(method, format = "default")

{

    # Универсальный метод проверки корректности введенных данных

    repeat{

        switch(

            format,

            "inputName" = { input = readline(prompt = "Введите ваше имя: ")},

            "clearNumeric" = { input = sub(",", ".", readline(prompt = "Введите дробное число: ")) },

            "vehicleSpeed" = { input =  readline(prompt = "Введите скорость целым числом в км/ч: ") },

            "default" = { input = readline(prompt = "Введите строку: ") }

        )

        result <- method(input)

        if(!is.na(result)){

            return(result)

        }

    }

}

# Задание 1.1

task11 <- function(){

    name <- validate(as.character, "inputName")

    cat(sprintf("Hello, %s!", name))

}

# Задание 1.2

task12 <- function(){

    # Можно пытаться ввести даже строку, все равно прога не даст

    num1 <- validate(as.numeric, "clearNumeric")

    num2 <- validate(as.numeric, "clearNumeric")

    cat("Сумма чисел:", num1 + num2);

}

# Задание 1.3

task13 <- function(){

    spd <- validate(as.integer, "vehicleSpeed")

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

}

# ================================

#             Векторы

# ================================

# Задание 2.1

task21 <- 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')

}

# Задание 2.2

task22 <- function(){

    # Указываем произвольный вектор

    vector <- c(1, 200, 0, TRUE, 5, 6, 7, NA, 2, -1, 20.0, "TEST")

    # Заменяем элемент, индекс которого равен длине вектора

    vector[length(vector)] = NA

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

}

# Задание 2.3

task23 <- 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)))

}

# Задание 2.4

task24 <- function(){

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

# В отличие от предыдущего задания, здесь неэффективно использовать which

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

}

# Задание 2.5

task25 <- function(){

    amount = 100;

    vector <- c()

    for(c in 1:amount)

        vector[c]=c

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

}

# Задание 2.6

task26 <- function(){

    repeats = 5;        # Количество повторений

    yearFrom = 2000;    # Год начала отсчета

    # Генерируем страны

    countries = c()

    for(i in c("France", "Italy", "Spain"))

        countries <- c(countries, rep(i, repeats))

    # Генерируем года

    years = c();

    for(i in 1:repeats)

        years[i]=yearFrom+i-1

    # Выводим таблицу

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

    print(table)

}

# Задание 2.7

task27 <- function(){

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

    average = sum(income) / length(income)

    # Производим замену

    income\_class <- replace(temp <- replace(income, income < average, 0), temp >= average, 1)

    cat("Вектор income\_class:", income\_class)

}

# ================================

#          Запуск задания

# ================================

# Метод запуска заданияS

startTask <- function(arg){

    shell("cls");

    switch(

      arg,

      "1.1" = task11(),

      "1.2" = task12(),

      "1.3" = task13(),

      "2.1" = task21(),

      "2.2" = task22(),

      "2.3" = task23(),

      "2.4" = task24(),

      "2.5" = task25(),

      "2.6" = task26(),

      "2.7" = task27(),

    )

    cat('\n');

}

# Непосредственно запуск задания

# startTask("1.1")

# startTask("1.2")

# startTask("1.3")

# startTask("2.1")

# startTask("2.2")

# startTask("2.3")

# startTask("2.4")

# startTask("2.5")

# startTask("2.6")

# startTask("2.7")